AD-A080 807   SYSTEMS RESEARCH LABS INC DAYTON OHIO   F/G 14/2
DEVELOPMENT OF MTQ TRACKER MODEL AND IDENTIFICATION OF MODEL PA--ETC(U)
AUG 79 R S KOU, B C GLASS, M S MORAN   F33615-79-C-0500
UNCLASSIFIED SRL-6872-9   AMRL-TR-79-80   NL

END
DATE
FILMED
3 80
DDC

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (18) AMRL-TR-79-80 (19)	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) DEVELOPMENT OF MTQ TRACKER MODEL AND IDENTIFICATION OF MODEL PARAMETERS.		5. TYPE OF REPORT & PERIOD COVERED (9) Technical Report.
		6. PERFORMING ORG. REPORT NUMBER (14) SRL-6872-9
7. AUTHOR(s) (10) Robert S. Kou, Betty C. Glass, Martin S. Moran, Maris M. Vikmanis		8. CONTRACT OR GRANT NUMBER(s) (15) F33615-79-C-0500
9. PERFORMING ORGANIZATION NAME AND ADDRESS Systems Research Laboratories, Inc. 2800 Indian Ripple Road Dayton, Ohio 45440		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (17) (16) 62202F, 6893 04-33
11. CONTROLLING OFFICE NAME AND ADDRESS Aerospace Medical Research Laboratory, Aerospace Medical Division, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio 45433		12. REPORT DATE (11) August 1979
		13. NUMBER OF PAGES 87
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) (12) 86		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

MTQ, tracker model, parameter identification, observer theory, least-squares curve fitting, anti-aircraft artillery system.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This technical report summarizes the development of the observer model for the gunner's tracking performance in anti-aircraft artillery (AAA) systems. The Luenberger observer theory is used to design this gunner model which is composed of an identity observer, a feedback controller and a remnant element. The structure of the observer model is much simpler than that of the optimal control model developed by Kleinman, Baron and Levison. The computer execution time of the AAA system simulation using the observer model is about 15% of that → over

340 400 JB

using the optimal control model. Short execution time in the AAA system simulation is one of the main advantages of this model.

A parameter identification program based on the least squares curve-fitting method and the Gauss-Newton gradient algorithm is developed to identify systematically the parameter values of this gunner model. The ensemble mean and standard deviation of model predictions for both azimuth and elevation tracking errors are compared with human tracking data obtained from experiments conducted at the Aerospace Medical Research Laboratory, WPAFB, Ohio. Model predictions are in excellent agreement with the empirical data for several aircraft flyby and maneuvering trajectories. It is concluded that this gunner model and the parameter identification program can be used accurately and efficiently in the analysis of AAA effectiveness.

SUMMARY

This report describes the development of a mathematical model for gunner's tracking performance in MTQ Mode II tracking task which is a linear time-varying antiaircraft artillery system. The Luenberger observer theory is used to design the gunner model which is composed of three elements--a reduced-order observer, a feedback controller, and a remnant element. An important feature of the model is that its structure is simple, hence the computer simulation of man-in-the-loop AAA tracking systems using the gunner model requires only a short execution time. A parameter identification program based on the combined least squares curve-fitting method and the modified Gauss Newton gradient algorithm is developed to determine parameters of the model systematically. Model predictions of both azimuth and elevation tracking errors for several target flyby and maneuvering trajectories are shown to be in excellent agreement with the empirical data obtained from manned AAA simulation experiments conducted at the Aerospace Medical Research Laboratory, Wright-Patterson AFB, Ohio. It is concluded that the antiaircraft gunner model based on the observer theory can be accurately and efficiently used to study AAA weapon effectiveness and aircraft survivability.

PREFACE

This report documents a study performed by Systems Research Laboratories, Inc. (SRL), Dayton, Ohio, for the Aerospace Medical Research Laboratory (AMRL), Manned-Systems Effectiveness Division, Manned Threat Quantification program. This work was performed under Contract F33615-76-C-5001. The Contract Monitor was Mr. Robert E. Van Patten, and the Technical Manager was Capt. George J. Valentino. The SRL Project Manager was Mr. Charles McKeag.

The authors wish to extend their appreciation to Dr. Carroll N. Day, Mr. Walt Summers and Dr. Dan Repperger of Manned-Systems Effectiveness Division of the Aerospace Medical Research Laboratory, WPAFB, for many valuable discussions and helpful comments.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (cont.)

SECTION I

INTRODUCTION

The Luenberger observer theory [1] has been applied to design a gunner model by the authors in [2] through [8] which describes the gunner's tracking response in antiaircraft artillery (AAA) systems. The main highlights of the model are a simple model structure and accurate predictions of tracking errors. In [4], it has been pointed out that in the computer simulation of man-in-the-loop AAA tracking systems tremendous computer execution time can be saved by using the human operator (or gunner) model based on the observer theory rather than the optimal control model [9]. This is the advantage of a model with simple structure. The designed model has been applied to study the weapon effectiveness of linear time-invariant AAA tracking systems at the Aerospace Medical Research Laboratory of Wright-Patterson AFB, Ohio. This report will present a more general antiaircraft gunner model which can represent the gunner's tracking performance in the MTQ System which is a linear time-varying AAA tracking system. The basic structure of the model contains three elements--a reduced-order observer, a feedback controller, and a remnant element. However, the reduced-order observer is now a linear time-varying system. So the antiaircraft gunner model is also a linear time-varying system. A parameter identification program is developed to determine model parameters such that the model output can represent gunner's tracking function. The combined least squares curve-fitting method and the modified Gauss Newton gradient iterative algorithm are used to design the parameter identification

program. The parameter values of the gunner model are obtained by iteratively adjusting their values until the model predictions of azimuth and elevation tracking errors match the empirical data of the manned MTQ simulation experiments. A computer simulation program for the closed-loop MTQ tracking task is also developed. Simulation results are compared with the experimental data. Model predictions of tracking errors are shown to be accurate representations of actual gunner tracking data. Many figures showing computer simulation results for various flyby and maneuvering trajectories are also included in this report.

SECTION II

MTQ TRACKING SYSTEM

Figure 1 shows the block diagram of the MTQ tracking system. The visual display provides the information of tracking error e_T on a two dimensional visual device to the human operator (gunner). The tracking error e_T is the difference between the target angle Θ_T and the gunsight angle Θ_g. The x-axis and the y-axis components of the signal on the display represent the azimuth and the elevation tracking errors. The function of the human operator (gunner) in Figure 1 is to align the gunsight angle Θ_g to the target angle Θ_T. In the MTQ system, only one gunner performs angle tracking task (i.e., he tracks both azimuth and elevation angles). In the following analysis, the MTQ tracking system is decomposed into elevation and azimuth tracking systems as shown in Figures 2 and 3. The gunner models in these figures are mathematical models representing gunner's azimuth and elevation tracking responses to be designed in the next section. In Figure 3, the factor $\cos(\Theta_g)_{EL}$ where $(\Theta_g)_{EL}$ is the elevation gunsight output, is included in the modeling of the azimuth visual display for some optical consideration (see Reference [10] for detail). Therefore, the elevation and the azimuth tracking systems are coupled in the sense that the elevation gunsight output $(\Theta_g)_{EL}$ is used as one of the inputs to the azimuth tracking system.

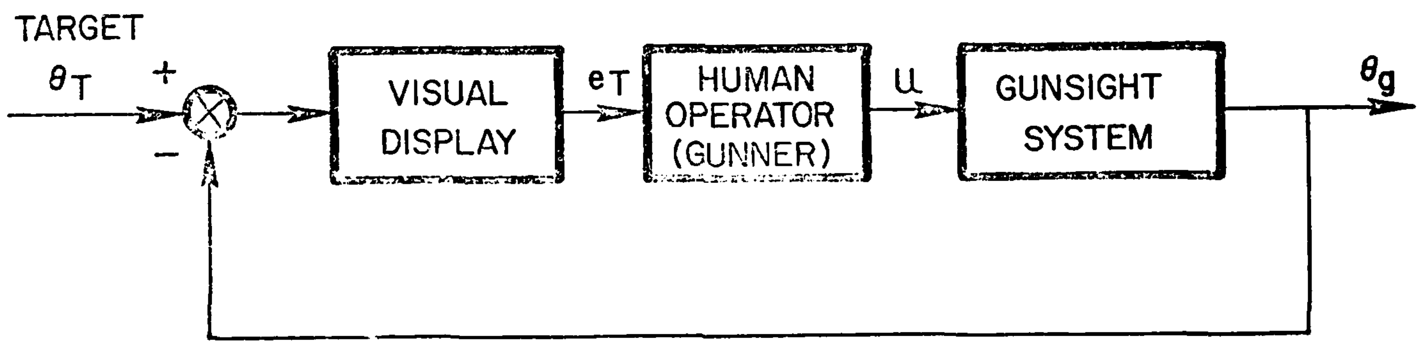


FIGURE I: BLOCK DIAGRAM OF AN AAA TRACKING SYSTEM

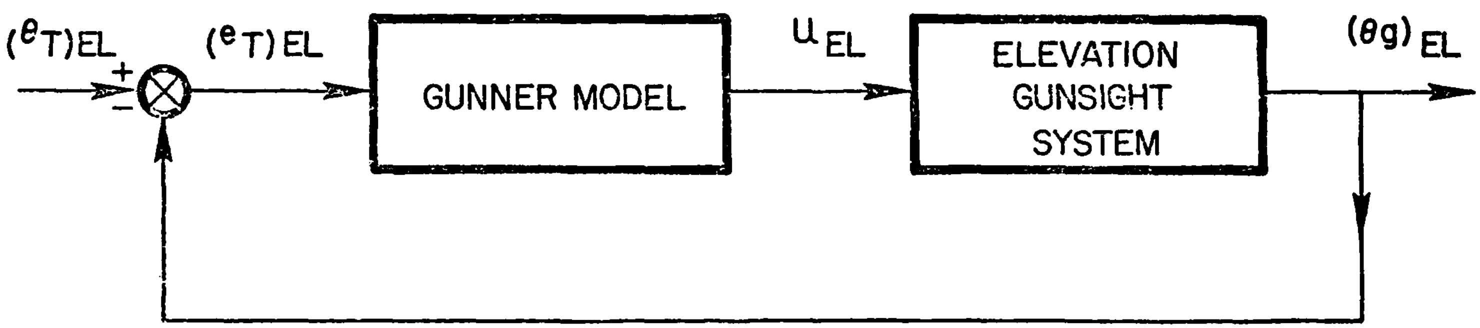


FIGURE 2: BLOCK DIAGRAM OF ELEVATION TRACKING SYSTEM

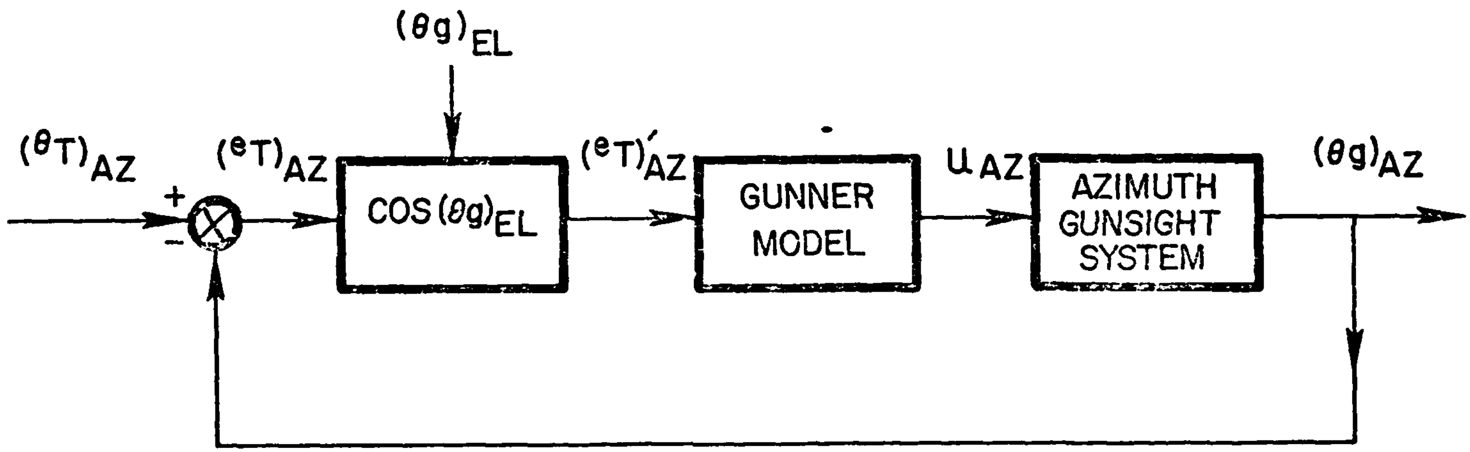


FIGURE 3: BLOCK DIAGRAM OF AZIMUTH TRACKING SYSTEM

The input-output relations of the azimuth and the elevation gunsight systems are represented by

$$(\dot{\Theta}_g)_{AZ} = 1.28u_{AZ} \tag{1}$$

and

$$(\dot{\Theta}_g)_{EL} = 1.34u_{EL} \tag{2}$$

Several simulated flyby and maneuvering trajectories of the target aircraft described in [10] are used as inputs to the AAA tracking system for simulation purposes. Now, a general state space equation of the azimuth or the elevation AAA tracking system including the gunsight system and the target motion can be derived in the following. Let us introduce state variables

$$x_{i1}(t) \stackrel{\Delta}{=} (e_T)_i = (\Theta_T)_i - (\Theta_g)_i$$

$$x_{i2}(t) \stackrel{\Delta}{=} (\dot{\Theta}_T)_i$$

where i = AZ or EL represents azimuth or elevation components of signals in the AAA tracking system. For example, x_{AZ1} and x_{AZ2} denote the azimuth components of tracking error and target angle rate respectively. And let $\underline{x}_i = [x_{i1}(t) \quad x_{i2}(t)]^T$ be the state vector. The general system dynamic equation and the measurement equation of the azimuth or elevation AAA tracking system can be derived by using Equations (1) and (2) and Figures 2 and 3.

$$\underline{\dot{x}}_i = A_i \underline{x}_k + B_i u_i = F_i (\ddot{\Theta}_T)_i \tag{3}$$

and

$$y_i = C_i \underline{x}_i \tag{4}$$

where i = AZ or EL,

$$A_i = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \qquad F_i = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

$$B_i \overset{\Delta}{=} \begin{bmatrix} b_{i1} \\ 0 \end{bmatrix} \quad \text{with } b_{AZ1} = -1.28 \text{ and } b_{EL1} = -1.34, \tag{5}$$

$$C_i \overset{\Delta}{=} [c_{i1} \; 0] \text{ with } c_{AZ1} = \cos\,(\Theta_g)_{EL} \text{ and } c_{EL1} = 1, \tag{6}$$

and u_i, $(\ddot{\Theta}_T)_i$ and y_i denote azimuth or elevation components of the gunner's control output, the target acceleration and the observed tracking error respectively. The above matrices A_i and F_i have the same values for azimuth or elevation cases but the vector B_i in Equation (5) shows different component values. Equation (6) shows that c_{AZ1} is time-varying while c_{EL1} is a constant equal to one. Therefore, the azimuth MTQ tracking system is a linear time-varying system while the elevation MTQ tracking system is only a linear time-invariant system. In the next section gunner models will be designed representing gunner's azimuth and elevation tracking responses corresponding to equations (3) and (4).

SECTION III

MTQ TRACKER MODEL DEVELOPMENT

The authors have developed an antiaircraft gunner model [2] through [8] based on the Luenberger reduced-order observer theory [1]. The basic structure of the model is shown in Figure 4 consisting of three main elements - a reduced-order observer, a state variable feedback controller, and a remnant element. The first two elements contain the deterministic part of the model while the remnant element represents the random (or stochastic) part of the model. The reduced-order observer gives an estimate of those unmeasurable state variables, and it is one of the characteristic capabilities of a human tracking. The state variable feedback controller represents the gunner's control function. All the effects of the various randomness sources due to human psychophysical limitations and of the modeling errors are equivalently lumped into one remnant element. These effects include the observation error, the neuromotor noise, target uncertainty modeling error, tracking error due to tracking difficulty with respect to various trajectories, etc. In this paper, more general mathematical equations of the model will be derived corresponding to the linear time-varying tracking system described by Equations (3) and (4).

Since the gunner doesn't have precise information about the target dynamics (i.e., the so-called human's uncertainty about target motion), the term representing target angle acceleration, $\ddot{\Theta}_T$, in Equation (3) will not be included in the design of the reduced-order observer of the gunner

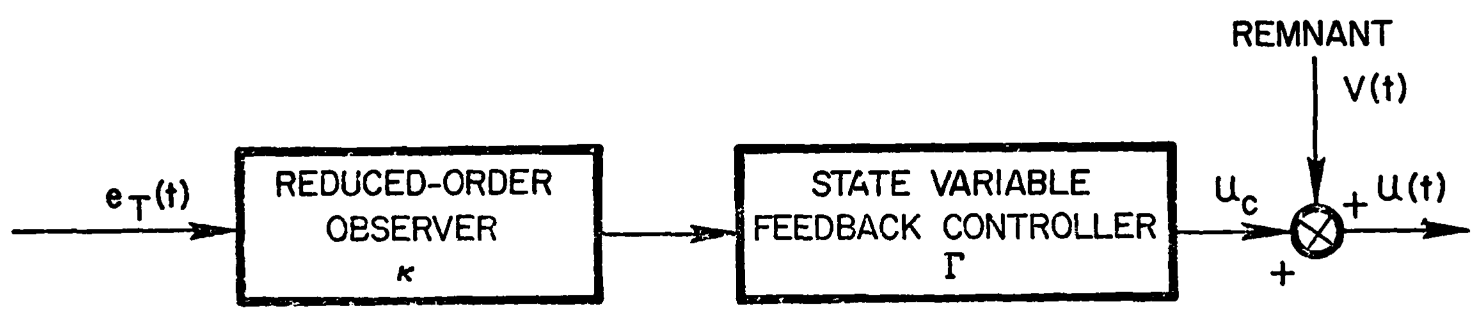


FIGURE 4: STRUCTURE OF THE ANTI-AIRCRAFT GUNNER MODEL

model. Let us assume that the gunner's understanding (or internal model) about the tracking system is described by

$$\dot{\underline{x}}_i = A_i \underline{x}_i + B_i u_i$$

and

$$y_i = C_i \underline{x}_i$$

For the convenience of designing a reduced-order observer, let us introduce a new state vectoı $\underline{x}_i'$

$$\underline{x}_i' = \begin{bmatrix} c_{i1} x_{i1} \\ \\ \\ x_{i2} \end{bmatrix}$$

where x_{i1} and x_{i2} are state components of $\underline{x}_i$, and c_{i1} is the first element of C_i of Equation (6). Then the equations representing the gunner's internal model about the tracking system can be rewritten as

$$\dot{\underline{x}}_i' = A_i' \, x_i' + B_i' \, u_i \tag{7}$$

and

$$y_i = C_i' \underline{x}_i' \tag{8}$$

where

$$A_i' = \begin{bmatrix} \dot{c}_{i1}c_{i1}^{-1} & c_{i1} \\ 0 & 0 \end{bmatrix} \quad B_i' = \begin{bmatrix} c_{i1}b_{i1} \\ 0 \end{bmatrix},$$

$$C_i' = [1 \quad 0]$$

Now C_1' is a constant vector. In fact, the new first state component of x_1' is measurable. Therefore, only the second state component x_{i2} of x_1' needs to be estimated in order to implement the state variable feedback controller. The reduced-order observer theory [1] is now applied to give an estimate $\hat{x}_{i2}$ of x_{i2} by using Equations (7) and (8). It can be shown that $\hat{x}_{i2}$ satisfies the following differential equation.

$$\dot{\hat{x}}_{i2} = -kc_{i1}\hat{x}_{i2} + k\dot{y} - k\,\dot{c}_{i1}c_{i1}^{-1}\,y - k\,c_{i1}b_{i1}u_c$$

where c_{i1} and b_{i1} are defined in Equations (5) and (6), k is the observer gain, and u_c is a linear feedback control law of the form

$$u_c = -[\gamma_1 \;\; \gamma_2] \begin{bmatrix} y \\ \hat{x}_{i2} \end{bmatrix} \tag{9}$$

where γ_1 and γ_2 are two feedback control gains. The values of k, γ_1 and γ_2 will be shown later. In order to bypass the time derivative of y in Equation (9), let us introduce a new variable z,

$$z = \hat{x}_{12} - ky(t) \tag{10}$$

then the observer dynamics can be expressed by

$$\dot{z} = -kc_{i1}z - \left(k^2c_{i1} + k\dot{c}_{i1}c_{i1}^{-1}\right)y - k\, c_{i1}b_{i1}u_c \tag{11}$$

Next, the control output u(t) of the gunner model of Figure 4 can be expressed by

$$u(t) = u_c(t) + v(t) = -\left[\gamma_1 \;\; \gamma_2\right]\begin{bmatrix} y \\ \hat{x}_{12} \end{bmatrix} + v(t) \tag{12}$$

where the remnant v(t) is modeled as a white noise with mean zero and with covariance function as follows

$$E[v(t)\ v(\tau)] = \left(\alpha_1 + \alpha_2\ \hat{\dot{\theta}}_T^2\ (t) + \alpha_3\ \hat{\ddot{\theta}}_T^2\ (t)\right)\ \delta(t-\tau)$$

$$\text{for all } t \text{ and } \tau \tag{13}$$

where α_1, α_2, α_3, are three nonnegative parameters of the model to be determined, $\hat{\dot{\theta}}_T$ and $\hat{\ddot{\theta}}_T$ are estimated target angle rate and acceleration respectively. Note that by definition $\hat{\dot{\theta}}_T$ is x_{12} and can be obtained from

Equation (9) or Equation (11), $\hat{\dot{\theta}}_T$ can be obtained by taking a first order Taylor series expansion of $\hat{\dot{\theta}}_T$. Equations (11) through (13) are the mathematical equations of a general gunner model. By selecting i = AZ or EL, this model represents the gunner's response in azimuth or elevation tracking task respectively. Next the gunner model equations will be combined with Equations (3) and (4) of the tracking system (which includes the gunsight system and the target motion) to form overall equations of the man-in-the-loop AAA tracking system. For the following analysis, the state components of the overall system are selected to be

$$\underline{X}_i = \begin{bmatrix} y_i \\ x_{i2} - ky_i \\ x_{i2} - \hat{x}_{i2} \end{bmatrix} \tag{14}$$

where y_i, x_{i2}, and $\hat{x}_{i2}$ are defined in Equations (4), (3) and (9) and k is the observer gain. Then the state space equation of the overall system can be described by

$$\dot{\underline{X}}_i = \underline{A}_i\underline{X}_i + \underline{F}_i\left(\ddot{\theta}_T\right)_i + \underline{D}_i v_i \tag{15}$$

where $\underline{A}_i$, $\underline{F}_i$, and $\underline{D}_i$ matrices are

$$\underline{A}_i = \begin{bmatrix} \dot{c}_{i1}c_{i1}^{-1} + c_{i1}k - b_{i1}c_{i1}\left(\gamma_1 + \gamma_2 k\right) & c_{i1} - b_{i1}c_{i1}\gamma_2 & b_{i1}c_{i1}\gamma_2 \\ -c_{i1}k^2 - k\dot{c}_{i1}c_{i1}^{-1} + b_{i1}kc_{i1}\left(\gamma_1 + \gamma_2 k\right) & -c_{i1}k + b_{i1}\gamma_2 kc_{i1} & -b_{i1}\gamma_2 kc_{i1} \\ 0 & 0 & -kc_{i1} \end{bmatrix}$$

$$\underline{F}_i = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \qquad \underline{D}_i = \begin{bmatrix} b_{i1}c_{i1} \\ -b_{i1}kc_{i1} \\ -b_{i1}kc_{i1} \end{bmatrix}$$

and v_i is the remnant element in which the parameters α_1, α_2, and α_3 of its covariance function may have different values for azimuth and elevation tracking cases. The parameters of the gunner model, i.e., k, γ_1, γ_2, α_1, α_2, and α_3 of Equations (11), (12), and (13) will be determined in the next section.

Section IV

COMBINED CURVE-FITTING IDENTIFICATION PROGRAM

Before the gunner model based in the observer theory can be used to describe gunner's tracking response, the parameters k, γ_1, γ_2, α_1, α_2, α_3, of Equations (9) through (13) should be determined first. The authors have developed a curve-fitting parameter identification program using the least squares method [11] and the Gauss Newton gradient method [12]. Model predictions of the ensemble mean and the ensemble standard deviation of tracking errors will be used in the curve-fitting program. The MTQ elevation tracking system will be used an as example. The corresponding equations are derived in terms of model parameters in the following.

Taking expectation values of both sides of Equation (15), we have

$$\dot{\overline{X}}_{EL} = \underline{A}_{EL}\overline{X}_{EL} + \underline{F}_{EL}\left(\ddot{\theta}_T\right)_{EL} \tag{16}$$

where $\overline{X}_{EL} = E\ [\underline{X}_{EL}]$. The first component $\overline{x}_1$ of $\overline{X}_{EL}$ is the model prediction of ensemble mean of tracking error. By solving vector differential Equation (16), $\overline{x}_1$ can be expressed in terms of model parameters. $\left(\overline{x}_1(0)\right.$ is assumed to be zero.$\left.\right)$

$$\overline{x}_1(t) = \int_o^t \left[\frac{k + 1.34\gamma_1 + 1.34k\gamma_2}{1.34\gamma_1\left(k + 1.34\gamma_1\right)}\ e^{\ 1.34\gamma_1(t - \tau)} \ +\right.$$

$$\frac{1.34\gamma_2}{k + 1.34\gamma_1} \quad e^{-k(t - \tau)} \quad - \quad \frac{1 + 1.34\gamma_2}{1.34\gamma_1} \Bigg] \cdot \quad \ddot{\theta}_T(\tau) \; d\tau \tag{17}$$

Furthermore, let $P(t) = E\left[\left(\underline{X}_{EL}(t) - \overline{X}_{EL}(t)\right)\left(\underline{X}_{EL}(t) - \overline{X}_{EL}(t)\right)^T\right]$, and it can be shown that the first diagonal element p_{11} of P is the square of the ensemble standard deviation of tracking errors. It can be shown that P(t) satisfies the following covariance equation.

$$\dot{P} = \underline{A}_{EL} P + P \, \underline{A}^T{}_{EL} + \underline{D}_{EL} \left(\alpha_1 + \alpha_2 \, \hat{\dot{\theta}}{}^2_T(t) + \alpha_3 \, \hat{\theta}{}^2_T(t)\right) \underline{D}^T{}_{EL} \tag{18}$$

By solving this matrix differential equation and with the assumption $p_{11}(0) = 0$ the first diagonal element $p_{11}(t)$ of P(t) can be obtained.

$$p_{11}(t) = \int_o^t 1.34^2 \left\{ \frac{-1}{k + 1.34\gamma_1} \left[1.34k\gamma_2 \, e^{-k(t - \tau)} - \left(k + 1.34\gamma_1 + 1.34k\gamma_2\right) e^{1.34\gamma_1(t - \tau)} \right] \right\}^2 \cdot \left(\alpha_1 + \alpha_2 \, \hat{\dot{\theta}}{}^2_T(\tau) + \alpha_3 \, \hat{\theta}{}^2_T(\tau) d\tau\right) \tag{19}$$

Let

$$\bar{s}(t) = p_{11}(t)^{1/2} , \tag{20}$$

then $\bar{s}(t)$ denotes the model prediction of ensemble standard deviation of tracking errors. Equations (17) and (20) are ensemble mean and ensemble standard deviation of tracking errors which are explicit functions of model parameters.

The parameters of the antiaircraft gunner model will be determined systematically and simultaneously by a combined least squares curve-fitting identification program. The reference curves to be fitted in the curve-fitting program are obtained from empirical data of manned AAA simulation experiments conducted at the Aerospace Medical Research Laboratory of Wright-Patterson AFB, Ohio. Several simulated flyby and maneuvering aircraft trajectories of 35 seconds duration are used as target trajectories for the above experiments. Let $\bar{x}_1'(t)$ and $\bar{s}'(t)$ be the reference empirical mean and standard deviation of tracking errors which are obtained by averaging the results of twenty-five experimental simulation runs with the same target trajectory and the same subject. Now the criterion function J of the combined least squares curve-fitting program is defined as

$$J(\underline{a}) = \int_0^{t_f} \left[\left(x_1'(t) \quad \bar{x}_1(t,\underline{a}) \right)^2 + c \left(\bar{s}'(t) \quad \bar{s}(t,a) \right)^2 \right] dt \tag{21}$$

where t_f is the tracking duration (equal to 35 seconds in this case). $\bar{x}_1'$ and $\bar{s}'$ are empirical reference data, a is the parameter vector of the gunner model,

$$\underline{a}^T = \begin{bmatrix} k & \gamma_1 & \gamma_2 & \alpha_1 & \alpha_2 & \alpha_3 \end{bmatrix}$$

$\bar{x}_1$ and $\bar{s}$ are functions of time and parameter vector $\underline{a}$ as shown in Equations (17) through (20). c is a positive weighting factor selected to be one in this case. The criteron function J of Equation (21) can be considered as a combination of two integrations. One of them is an integration of the square of the error between the empirical mean tracking error $\bar{x}_1'$ and the model prediction of the ensemble mean tracking error $\bar{x}_1$. The other is a similar integration related to the standard deviation of tracking errors. Therefore, minimizing J with respect to the parameter vector $\underline{a}$ is called a combined least squares curve-fitting method. We would like to point out that in this method model parameters will be determined such that empirical mean and standard deviation of tracking errors are fitted simultaneously by the corresponding model prediction functions $\bar{x}_1$ and $\bar{s}$. Now the parameter identification task becomes a minimization problem, i.e., to find values of the parameter vector $\underline{a}$ which minimize the criterion function $J(\underline{a})$ of Equation (21). The modified Gauss Newton iterative gradient method is derived in Appendix A which iteratively adjusts parameter values to minimize the criterion function J. This iterative process will continue until the increments are smaller than a preassigned lower bound. A computer curve-fitting program is developed using the above mentioned methods and procedures. The highlights of the identification procedure are a fast method, accurate values and a systematic approach. Obviously, the parameter identification done by a computer is much faster than manual tuning. The combined least squares curve-fitting minimization procedure can definitely

provide more accurate parameter values than the trial-and-error approach. Finally, it is a systematic procedure to determine parameter values and can be applied to gunner models describing control responses for various AAA tracking systems. The parameter values of the gunner model determined by this identification procedure for both elevation and azimuth tracking tasks are listed in the following table.

Parameter / Gunner Model For	Observer Gain	Controller Gains		Coefficients of Remnant Covariance Function		
	k	γ_1	γ_2	α_1	α_2	α_3
Elevation Tracking	1.88	-1.99	-.745	.0000094	.025	.068
Azimuth	5.12	-3.51	-.762	.0000363	.00614	.0117

We would like to point out that the above parameter values are obtained consistently from the combined curve-fitting parameter identification program with various initial guesses. In addition, the parameter values of the gunner model depend on the dynamics of AAA gunsight systems. For different AAA gunsight dynamic systems, the combined curve-fitting program will determine different parameter values for the gunner model.

Section V

COMPUTER SIMULATION RESULTS

Once the parameters of the gunner model are determined, the gunner model is ready to be used in computer simulation of the AAA closed loop tracking system to describe gunner's tracking performance. These parameter values can be substituted into the elements of matrices $\underline{A}_i$, $\underline{F}_i$, and $\underline{D}_i$ of Equation (15) which is a mathematical dynamic model of the overall closed-loop AAA tracking system. In order to find a solution of Equation (15) without using convolution integrations, it can be discretized to be

$$\underline{X}_{n+1} = \phi \, \underline{X}_n + \Gamma_1 \, \ddot{\theta}_{T,n} + \Gamma_2 \, v_n \tag{22}$$

where

$$\underline{X}_{n+1} = \underline{X}\left(t_{n+1}\right) \quad \text{with } t_{n+1} = (n+1)\, \Delta t \text{ and } \Delta t = .066 \text{ seconds,}$$

$$\phi = \text{EXP}\left[\underline{A}_i \Delta t\right], \quad \Gamma_1 = \int_0^{\Delta t} \text{EXP}\left[\underline{A}_i \sigma\right] \, d\sigma \cdot \underline{F}_i \tag{23}$$

$$\Gamma_2 = \int_0^{\Delta t} \text{EXP}\left[\underline{A}_i \sigma\right] d\sigma \cdot \underline{D}_i, \quad \ddot{\theta}_{T,n} = \ddot{\theta}_T\,(t_n) \text{ and} \tag{24}$$

v_n is a random sequence with the following properties

$$E\left[v_n\right] = 0,$$

$$E\left[\left(v_n\right)\left(v_n\right)^T\right] = \frac{1}{\Delta t}\left(\alpha_1 + \alpha_2 \, \hat{\dot{\theta}}_T^2 \left(t_n\right) + \alpha_3 \, \hat{\ddot{\theta}}_T^2 \left(t_n\right)\right)$$

Taking expectation values of both sides of Equation (22), we get

$$\overline{X}_{n+1} = \phi \, \overline{X}_n + \Gamma_1 \, \ddot{\theta}_{T,n} \tag{25}$$

Let P_{n+1} denote the covariance of $\underline{X}_{n+1}$, then it can be shown that P_{n+1} satisfies the following matrix difference equation.

$$P_{n+1} = \phi \, P_n \, \phi^T + \Gamma_2 \frac{1}{\Delta t} \left(\alpha_1 + \alpha_2 \, \hat{\dot{\theta}}_T^2 \left(t_n\right) + \alpha_3 \, \hat{\ddot{\theta}}_T^2 \left(t_n\right) \right) \Gamma_2^T \tag{26}$$

Then the first element of $\overline{X}_{n+1}$ of Equation (25) and the square root of the first diagonal element of the matrix P_{n+1} of Equation (26) are the model predictions of the ensemble mean and the ensemble standard deviation of tracking errors respectively. Computer programs simulating MTQ closed loop tracking system are developed using the above recursive equations (25) and (26). The inputs to these programs are motion trajectories of target aircraft. Six flyby and maneuvering trajectories as shown in Figure 5 are used in this study. The antiaircraft weapon is located at the origin of local horizontal x-y plane and the z axis represents the altitude. The increment of each of the three axes is 1000 ft. A detailed description of the characteristics of these trajectories can be found in [10]. The output of the computer simulation program are ensemble mean and ensemble standard deviation of tracking errors which can be used to predict the gunner's tracking characteristics. Computer simulation results of the MTQ tracking system show that model predictions match well with the empirical data for both mean and standard deviation of

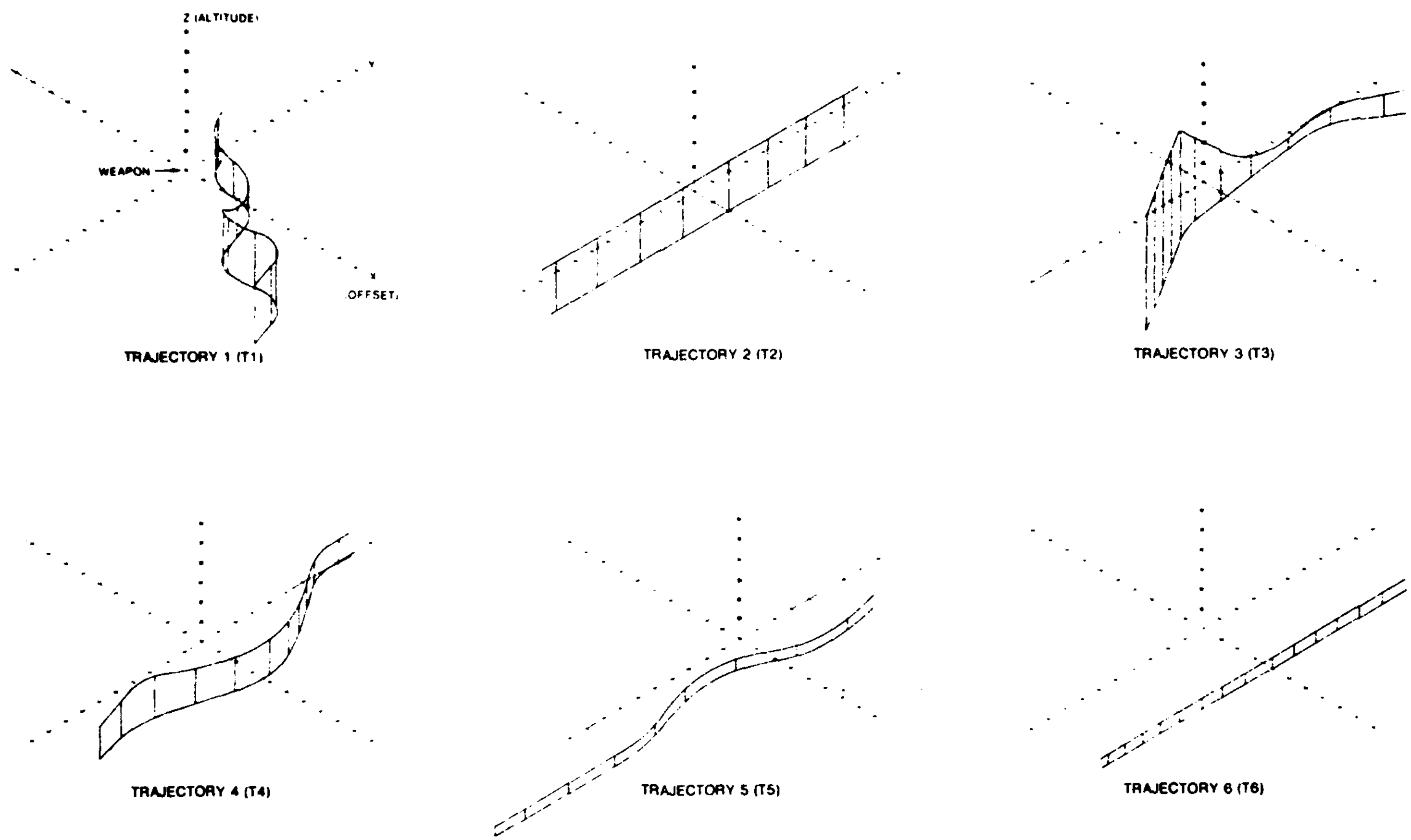


FIGURE 5. FLYBY AND MANEUVERING TARGET TRAJECTORIES

tracking errors. Simulation results are shown in Figures 6 through 29 for the trajectories of Figure 5. There are two curves on each of these figures. The solid curve is the empirical data which is obtained by averaging the results of twenty-five experimental simulation runs of 35 seconds duration. The dashed curve shows the ensemble value of the model prediction of the tracking error. Figures 6 through 9 show the results of elevation mean, elevation standard deviation, azimuth mean, and azimuth standard deviation of tracking errors for the maneuvering trajectory T1 of Figure 5. All of these four figures show that the MTQ tracker model can provide accuracy prediction of the corresponding empirical data. Next, Figures 10 through 13 show the similar results for the flyby trajectory T2 of Figure 5. Again three figures indicate the designed MTQ tracker model generating model predictions in excellent agreement with the empirical data for a flyby trajectory too. We would like to point out that the same set of parameter values determined by the combined curve-fitting identification program in the last section can be used to generate accurate model predictions for all six flyby and maneuvering trajectories of Figure 5. Therefore, it is concluded that for a given gunsight weapon system (e.g., MTQ) a same set of parameter values can be used in the observer model to predict human operator's tracking errors for all realistic simulated target trajectories. Hence, the observer model (i.e., MTQ tracker model) is a predictive model. Similarly, Figures 14 ~ 17, 18 ~ 21, 22 ~ 25, and 26 ~ 29 show the corresponding simulation results for trajectories T3, T4, T5, and T6. All these results verify that the designed observer model can describe

human's tracking response accurately. The names of these six trajectories in [10] are listed as follows for reference.

T1	Zigzag
T2	2 × 2 Flyby
T3	Fairpass
T4	RECON
T5	Weapon Delivery
T6	4 × .3 Flyby

A comparison between this model prediction and the optimal control model prediction [9] is shown in Figure 30. The dotted curve in Figure 30 is the optimal control model prediction of the tracking error. Obviously, it shows that the antiaircraft gunner model can represent the gunner's response as well as that by the optimal control model. However, the computer execution time by simulating the linear time-varying MTQ gun system using the gunner model is less than 15% of that using the optimal control model.

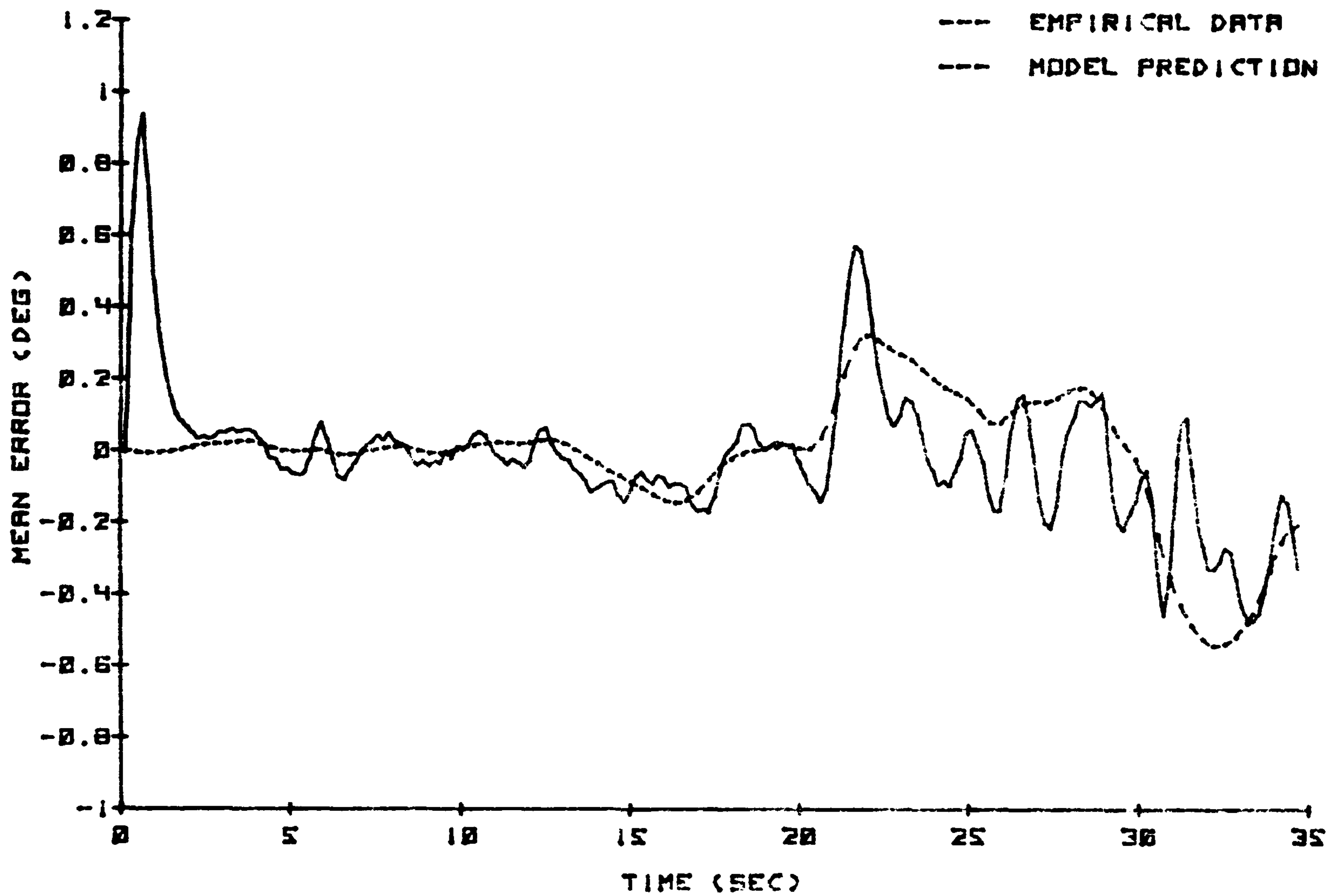


FIGURE 6 : MEAN TRACKING ERROR - ELEVATION T1

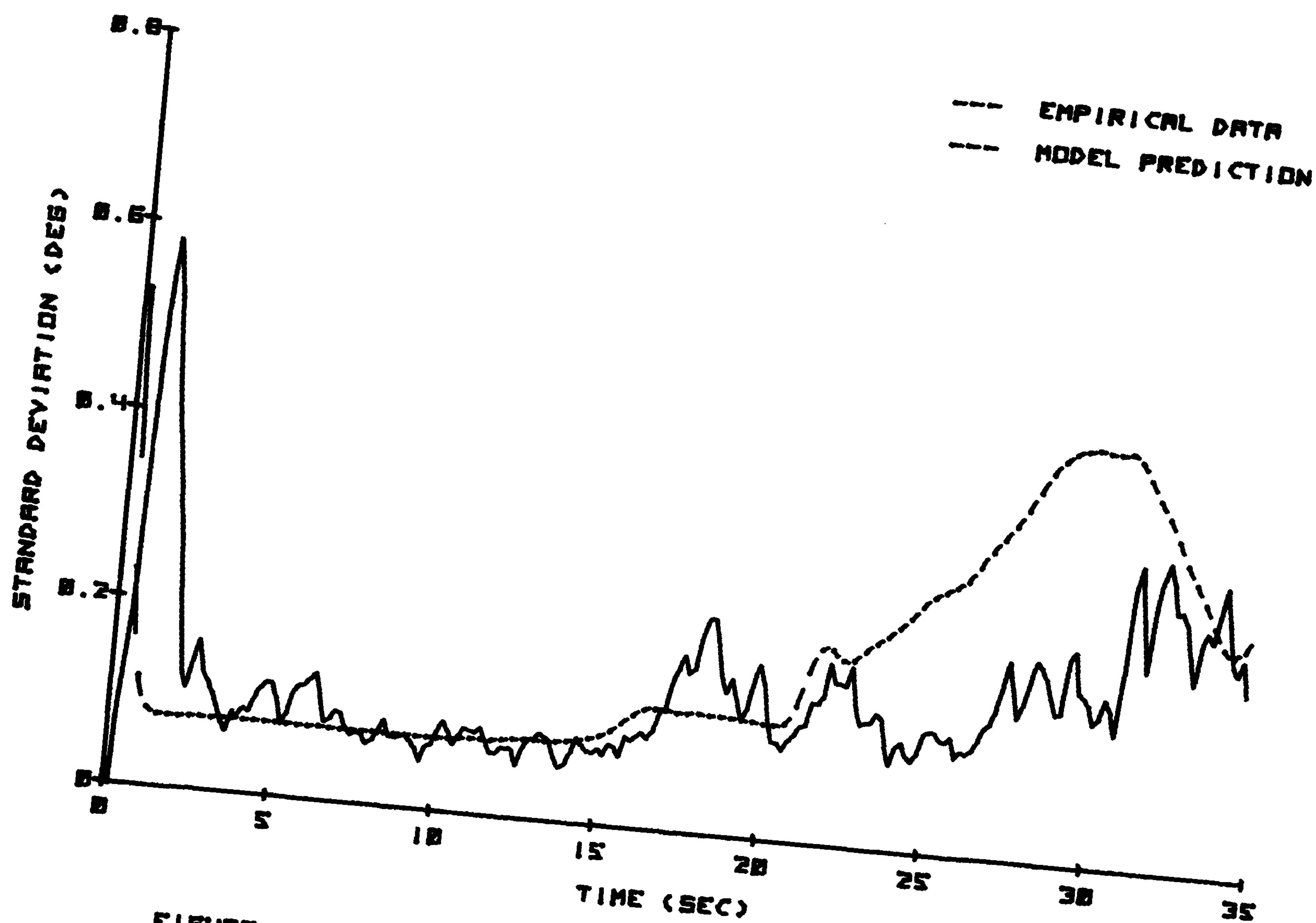


FIGURE 7 : STANDARD DEVIATION OF TRACKING ERROR - ELEVATION T1

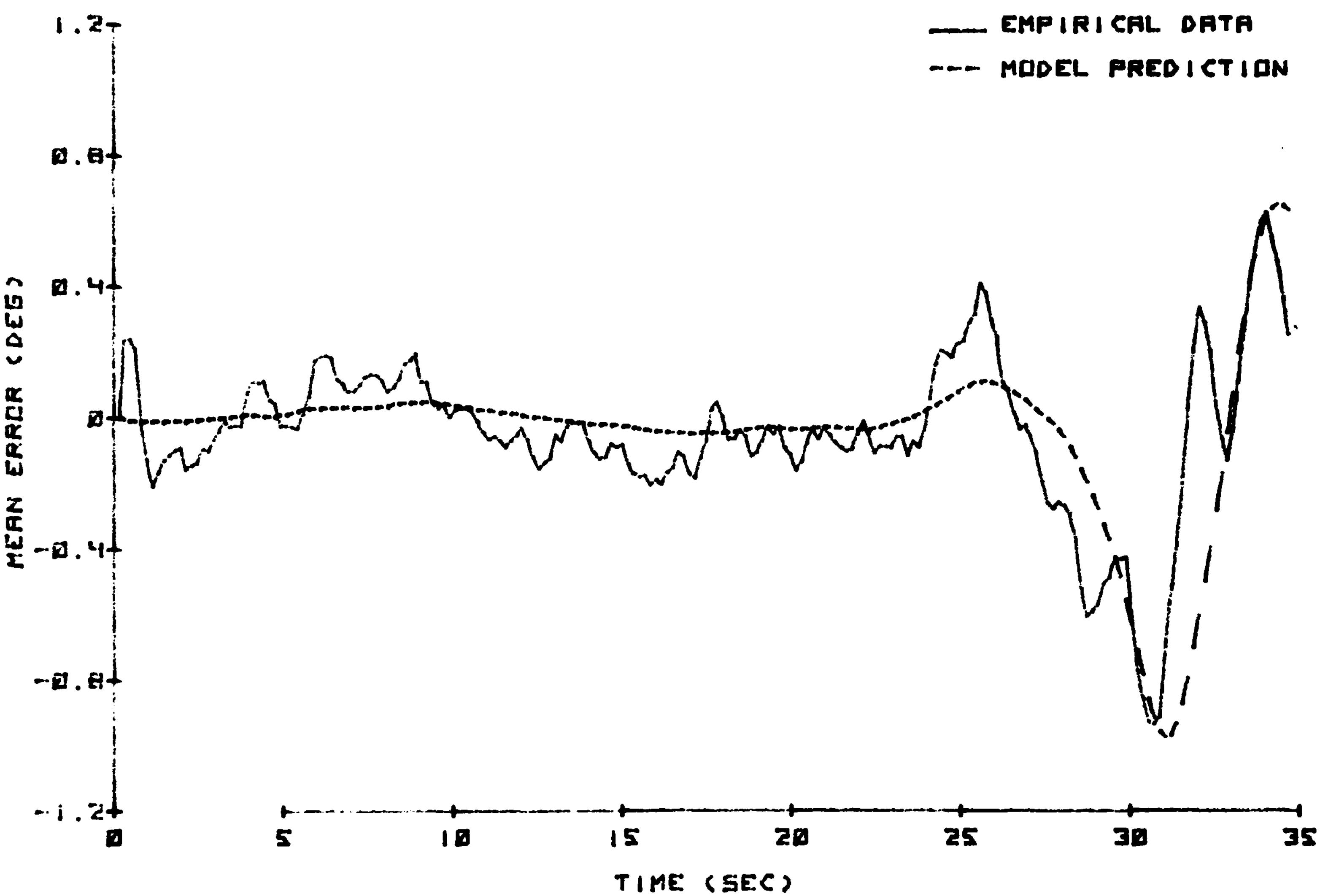


FIGURE 8 : MEAN TRACKING ERROR - AZIMUTH T1

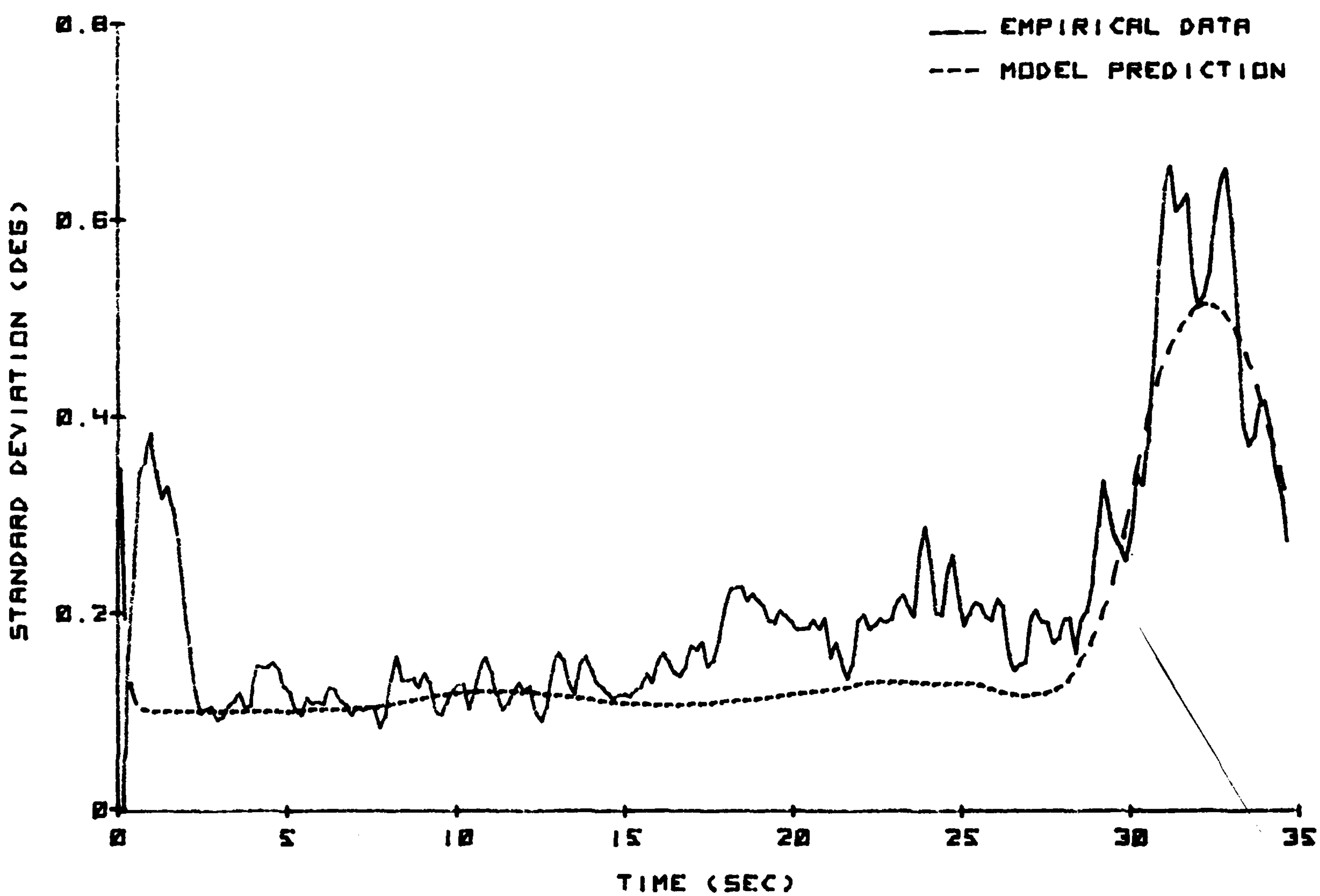


FIGURE 9 : STANDARD DEVIATION OF TRACKING ERROR - AZIMUTH T1

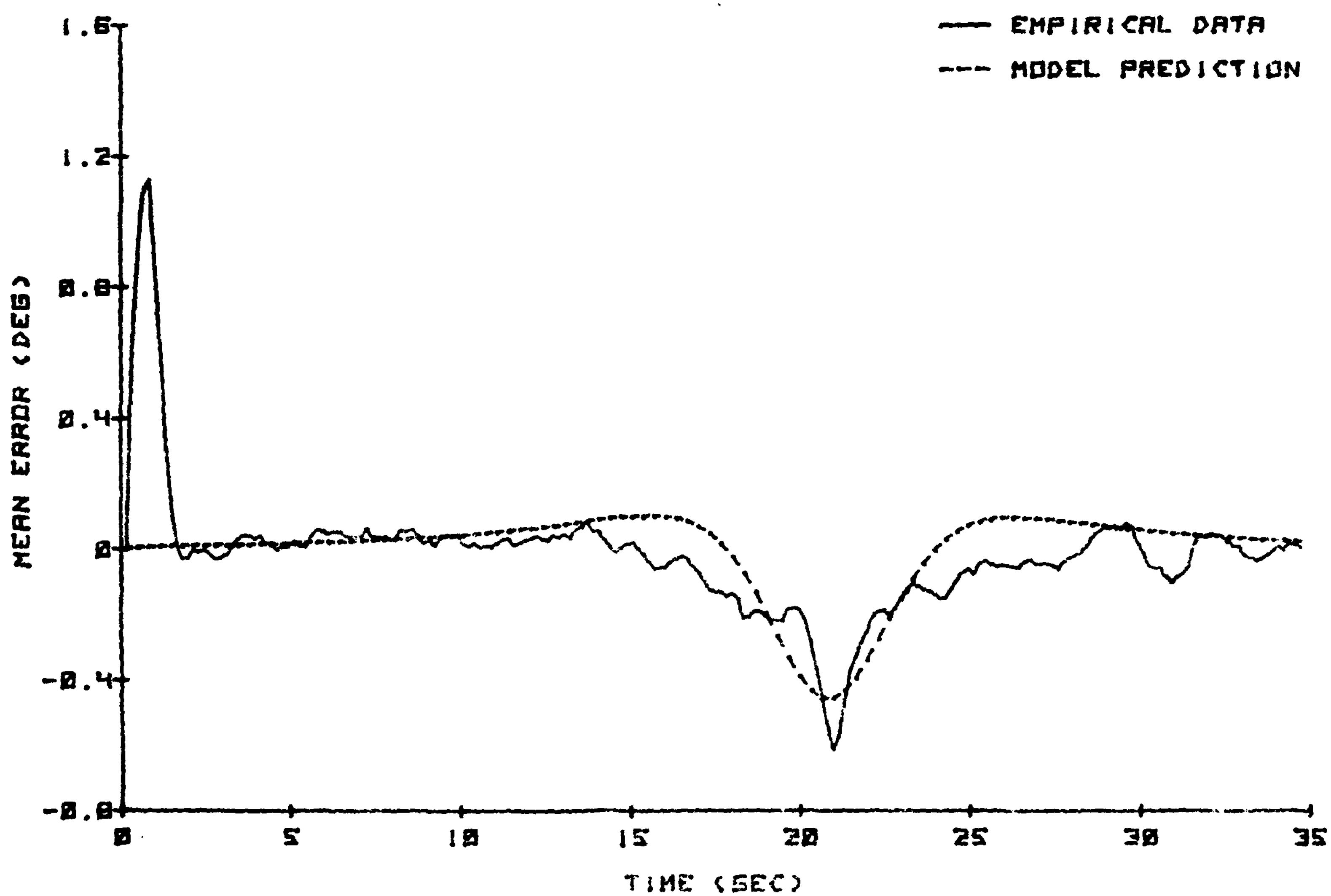


FIGURE 10 : MEAN TRACKING ERROR - ELEVATION T2

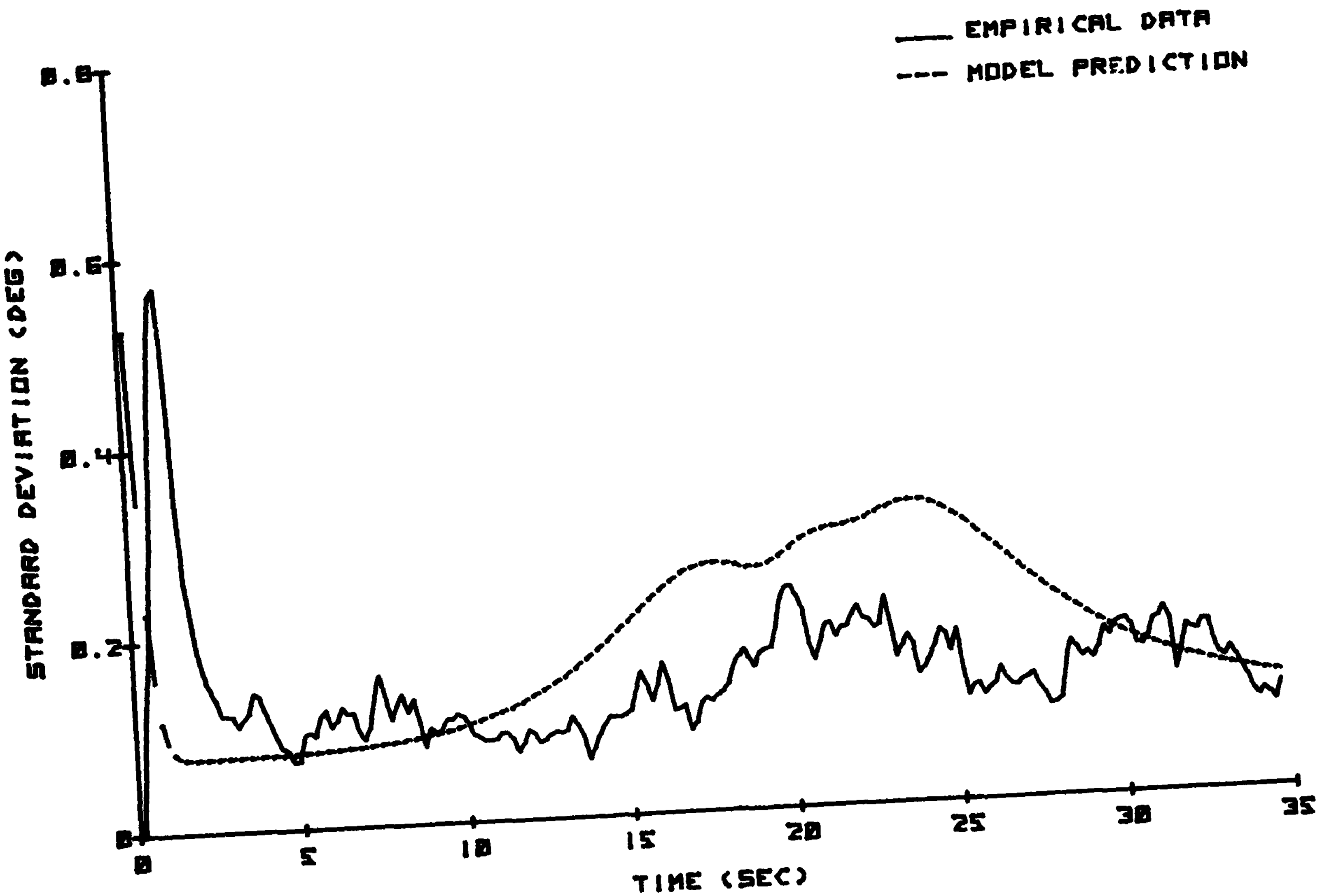


FIGURE 11: STANDARD DEVIATION OF TRACKING ERROR - ELEVATION T2

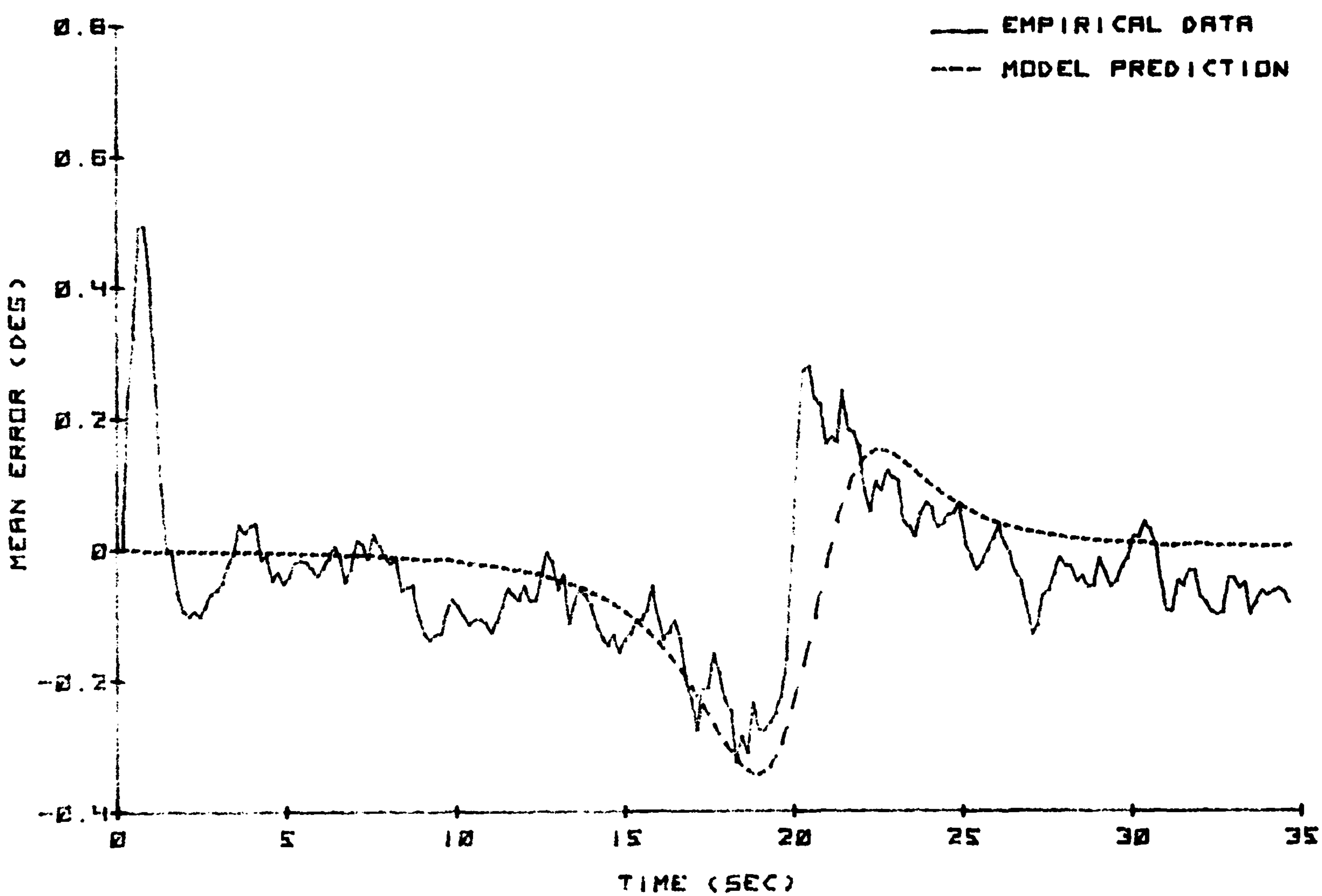


FIGURE 12: MEAN TRACKING ERROR - AZIMUTH T2

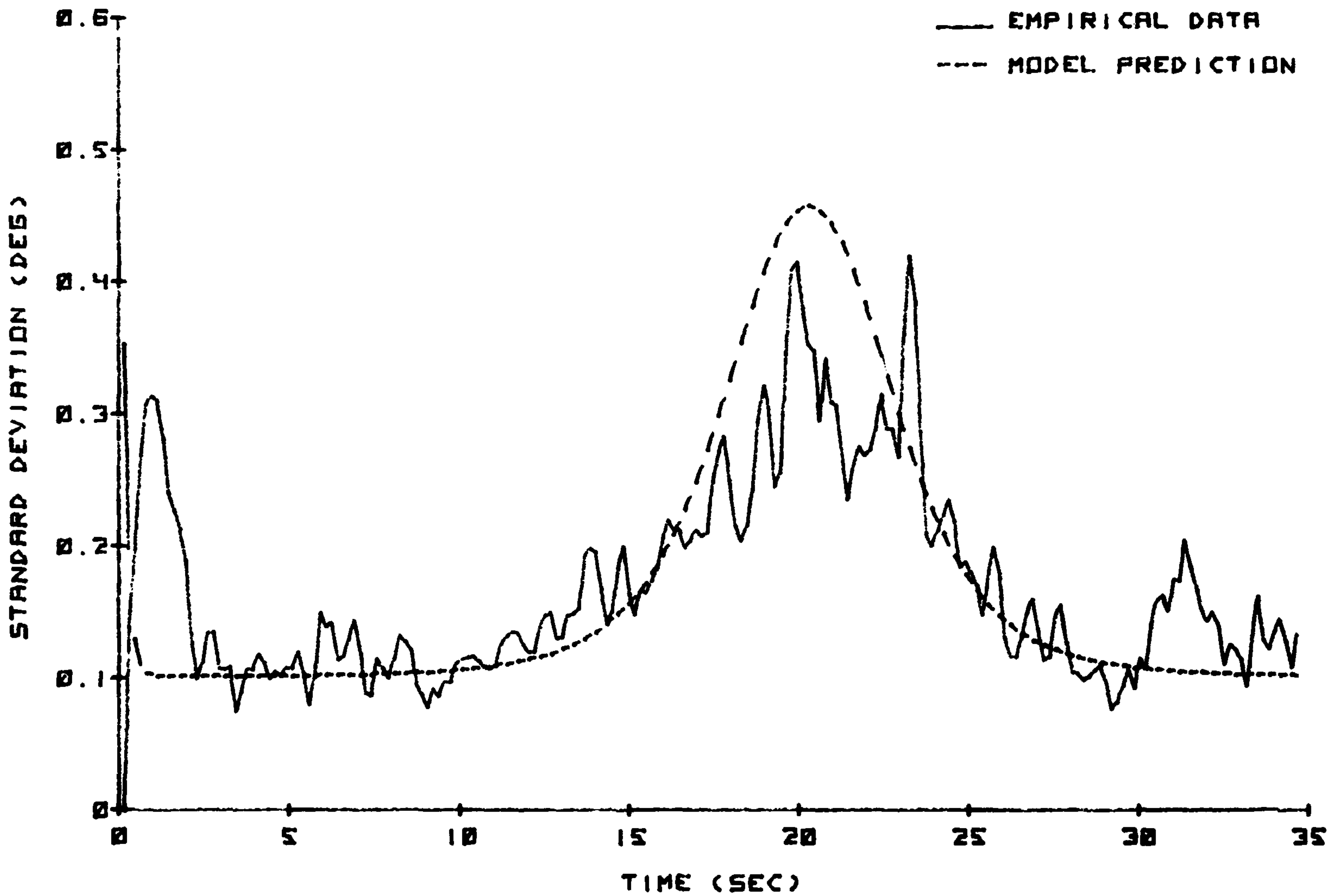


FIGURE 13 : STANDARD DEVIATION OF TRACKING ERROR - AZIMUTH T2

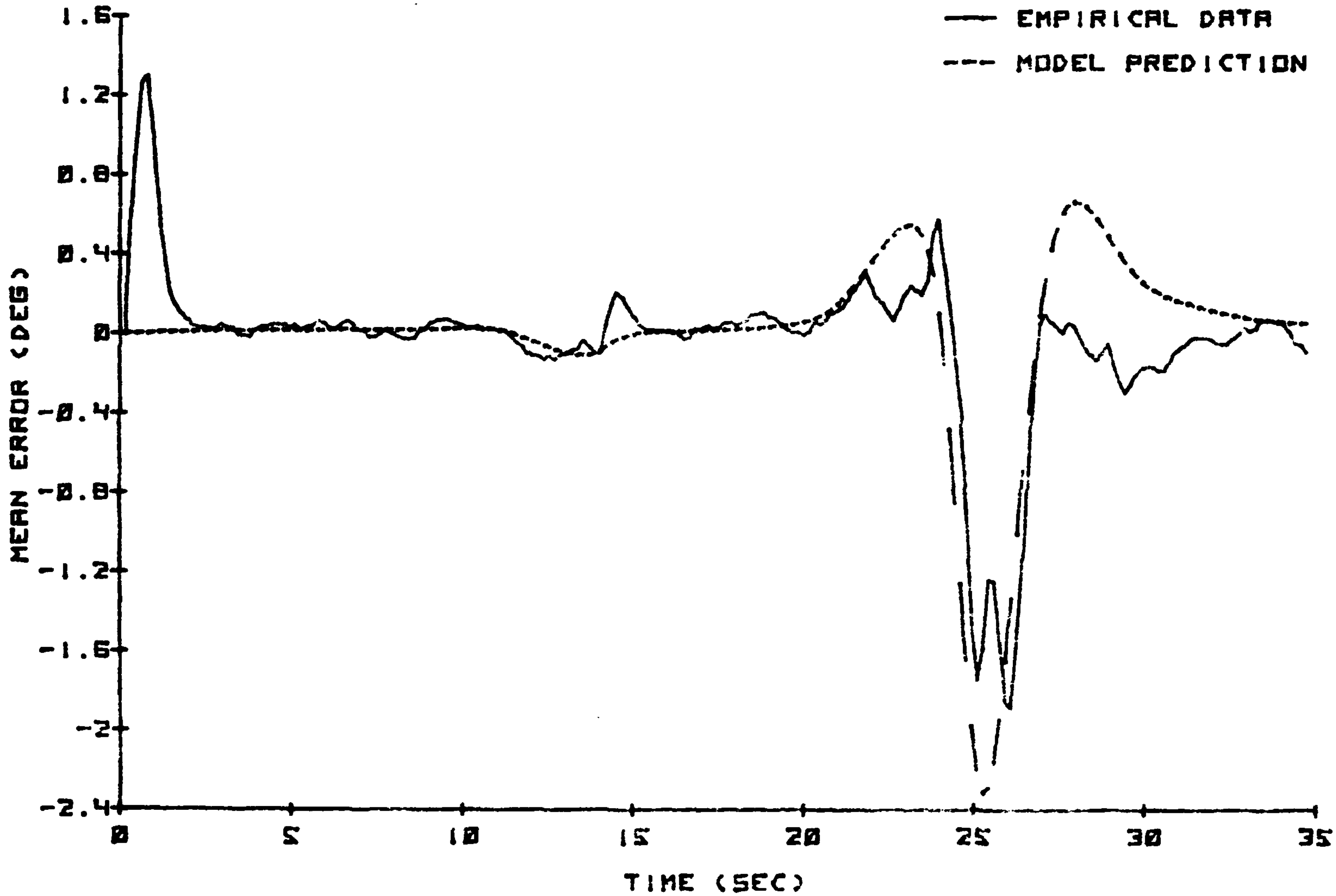


FIGURE 14 : MEAN TRACKING ERROR - ELEVATION T3

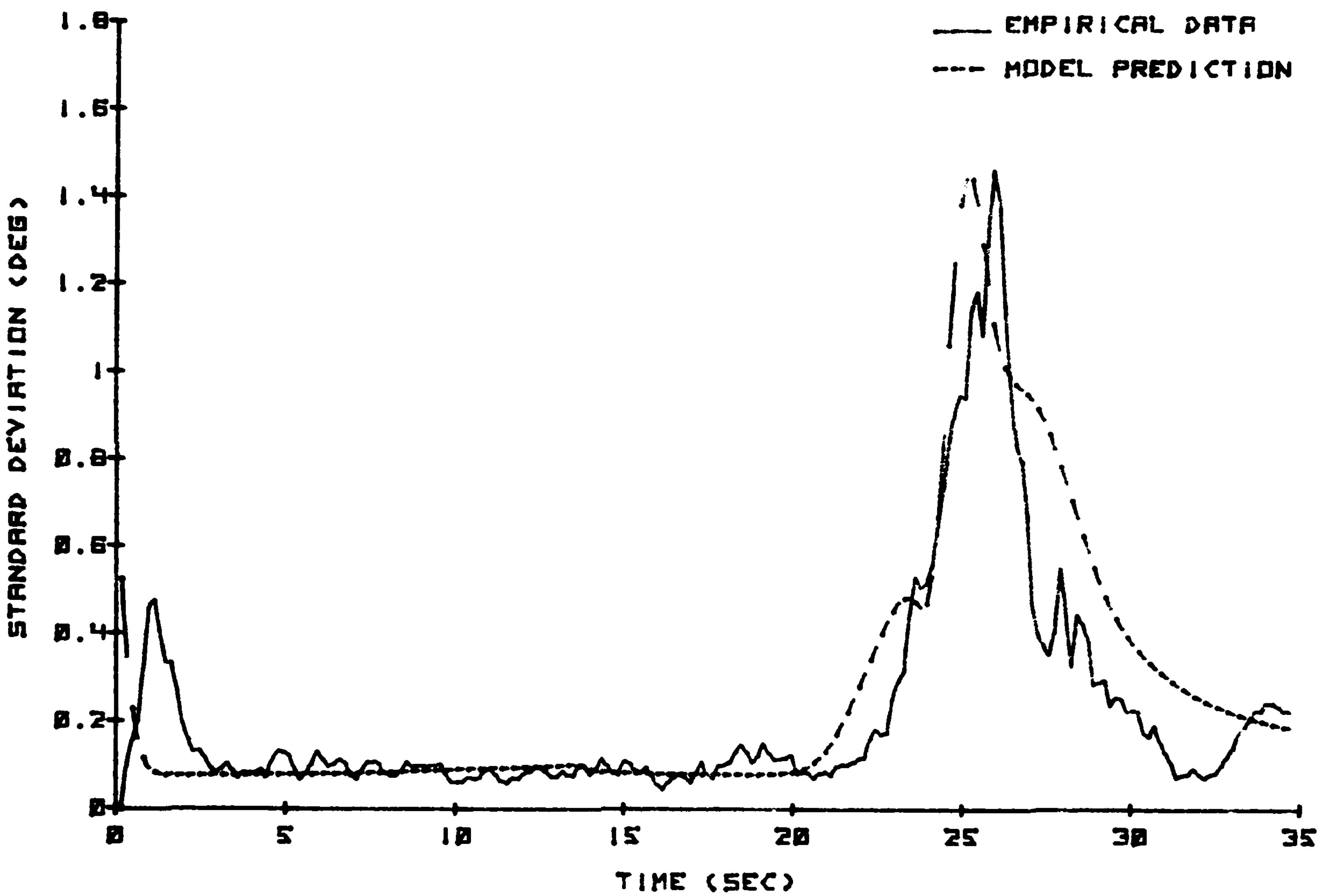


FIGURE 15: STANDARD DEVIATION OF TRACKING ERROR - ELEVATION T3

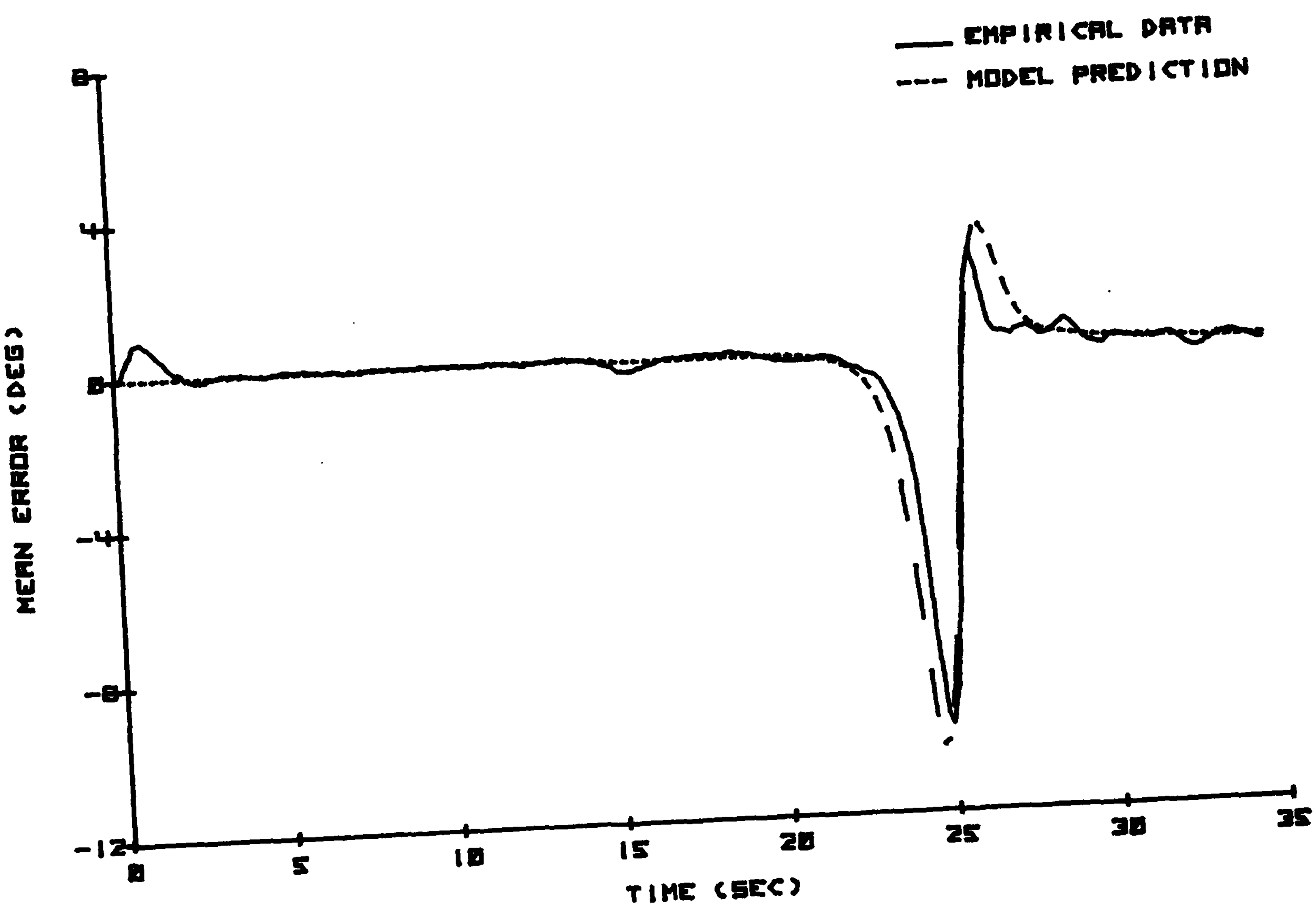


FIGURE 16 : MEAN TRACKING ERROR - AZIMUTH T3

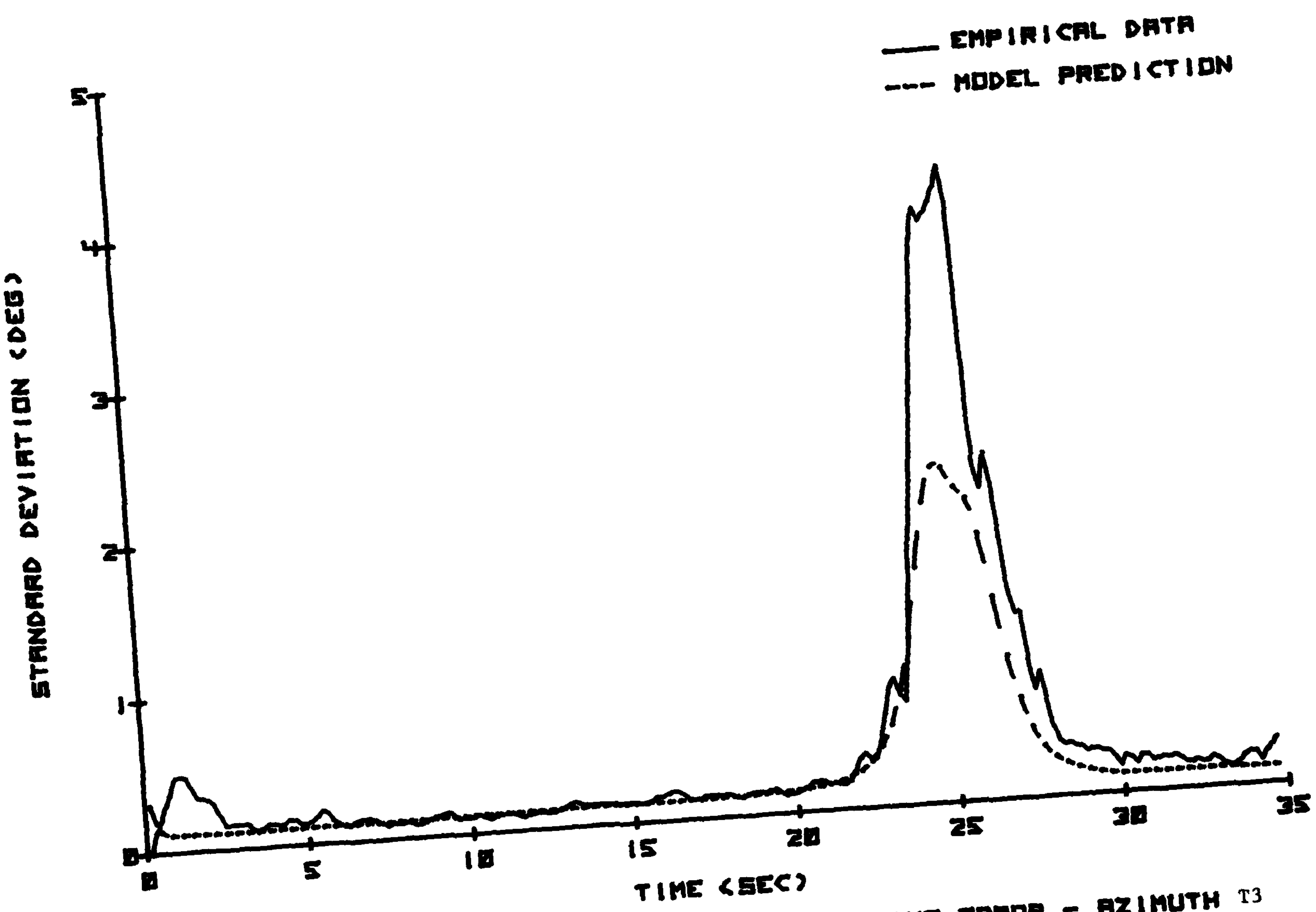


FIGURE 17: STANDARD DEVIATION OF TRACKING ERROR - AZIMUTH T3

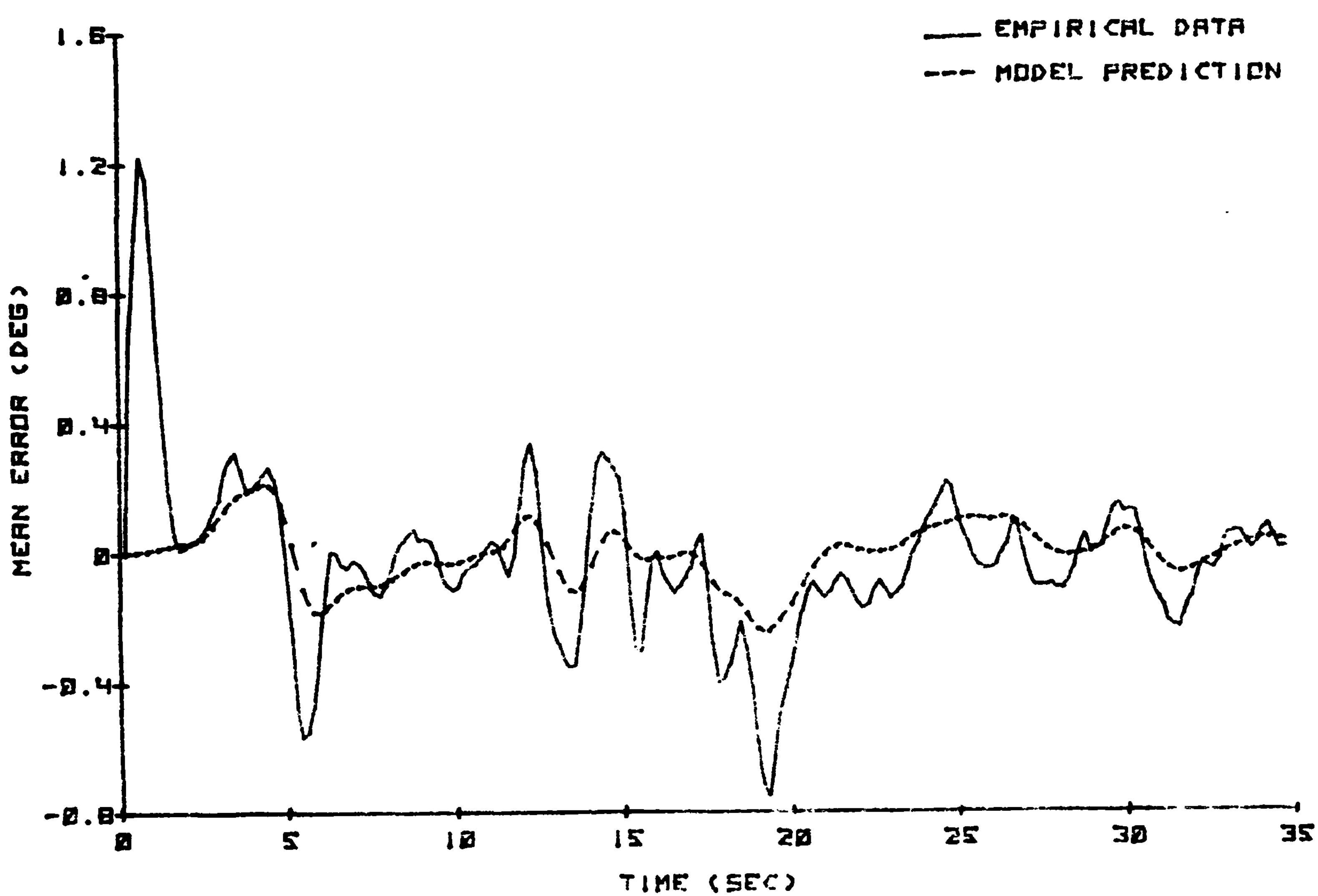
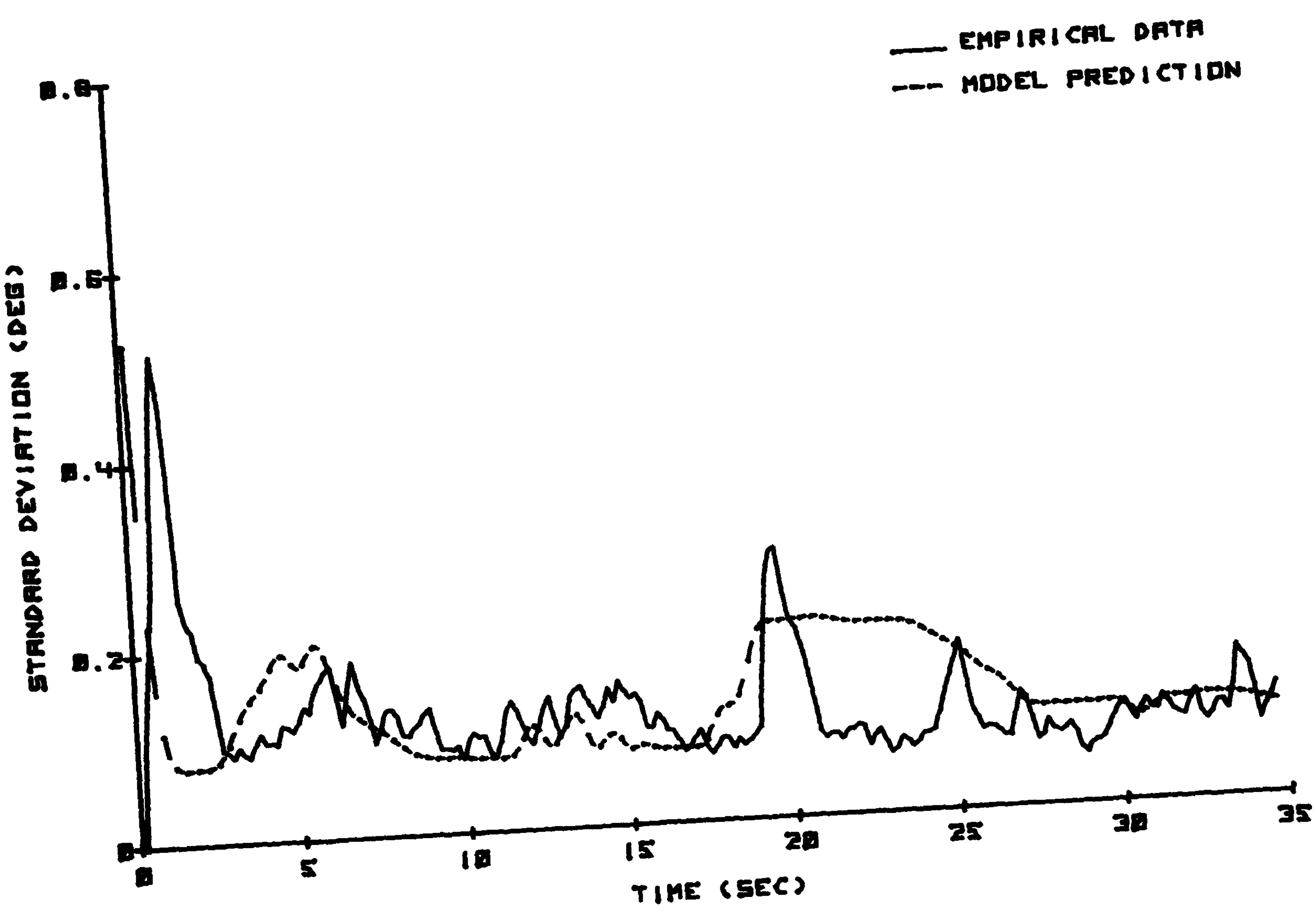


FIGURE 18 : MEAN TRACKING ERROR - ELEVATION T4



.FIGURE 19: STANDARD DEVIATION OF TRACKING ERROR - ELEVATION T4

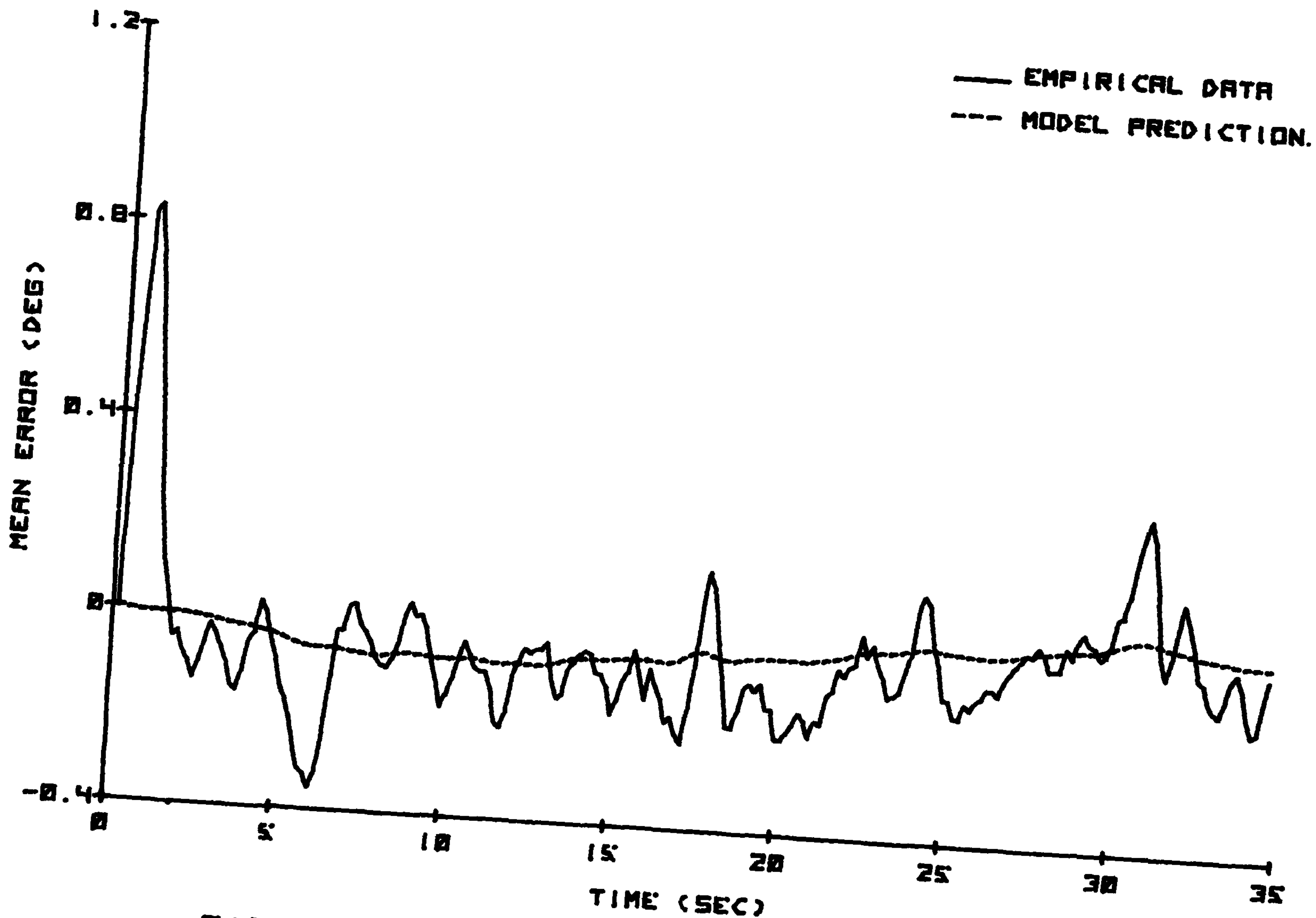


FIGURE 20 : MEAN TRACKING ERROR - AZIMUTH T4

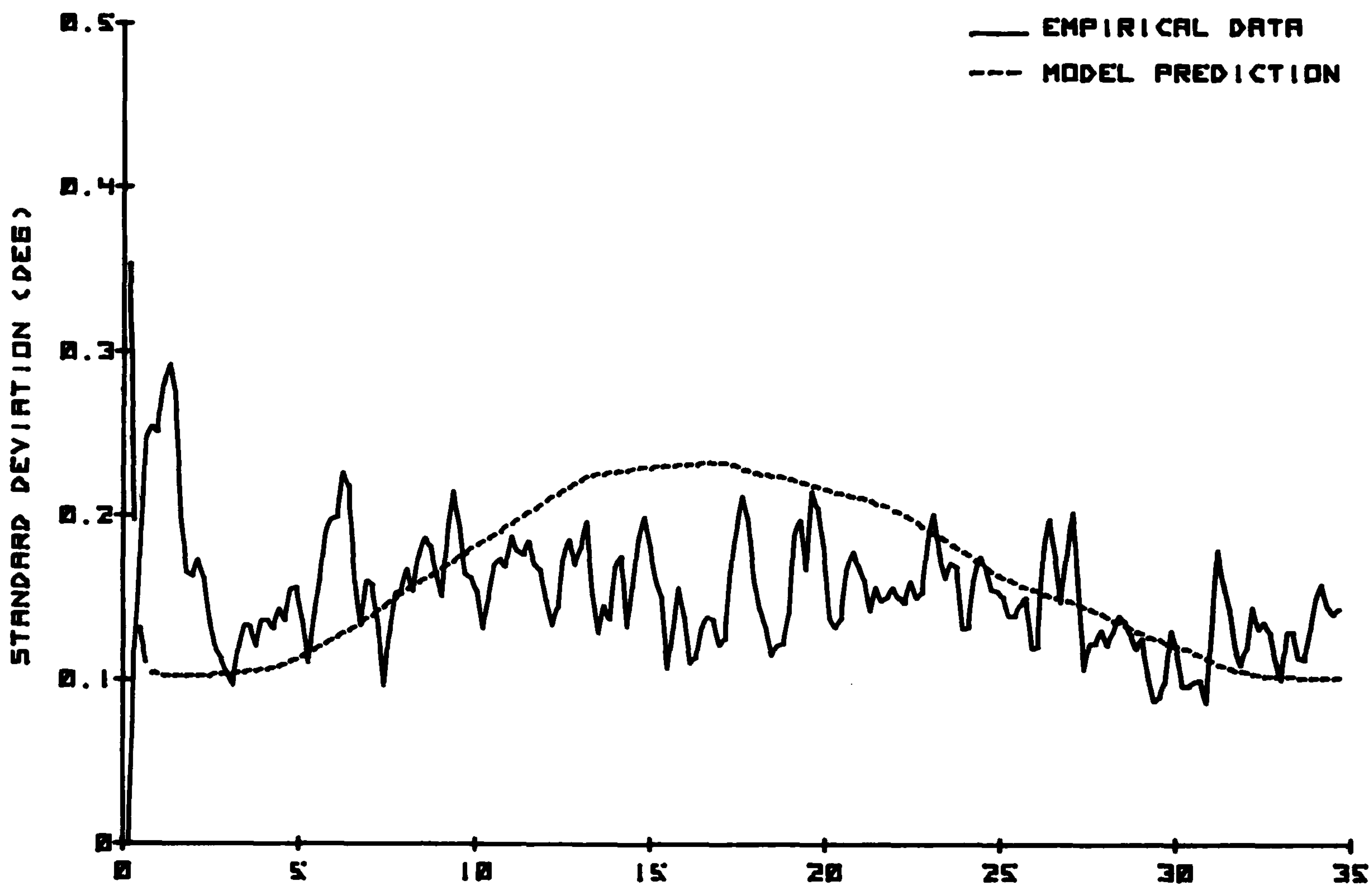


FIGURE 21 : STANDARD DEVIATION OF TRACKING ERROR - AZIMUTH T4

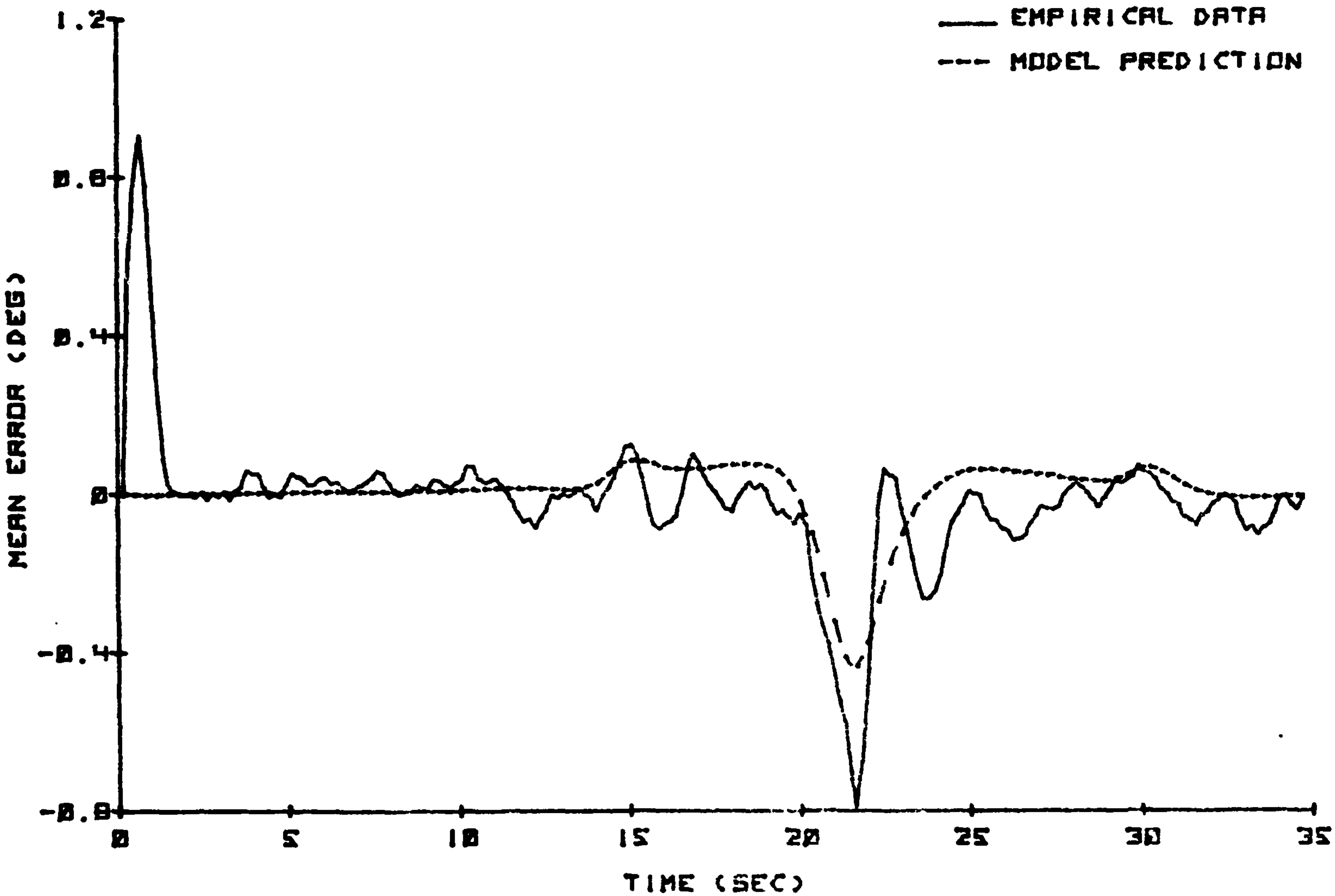


FIGURE 22: MEAN TRACKING ERROR - ELEVATION T5

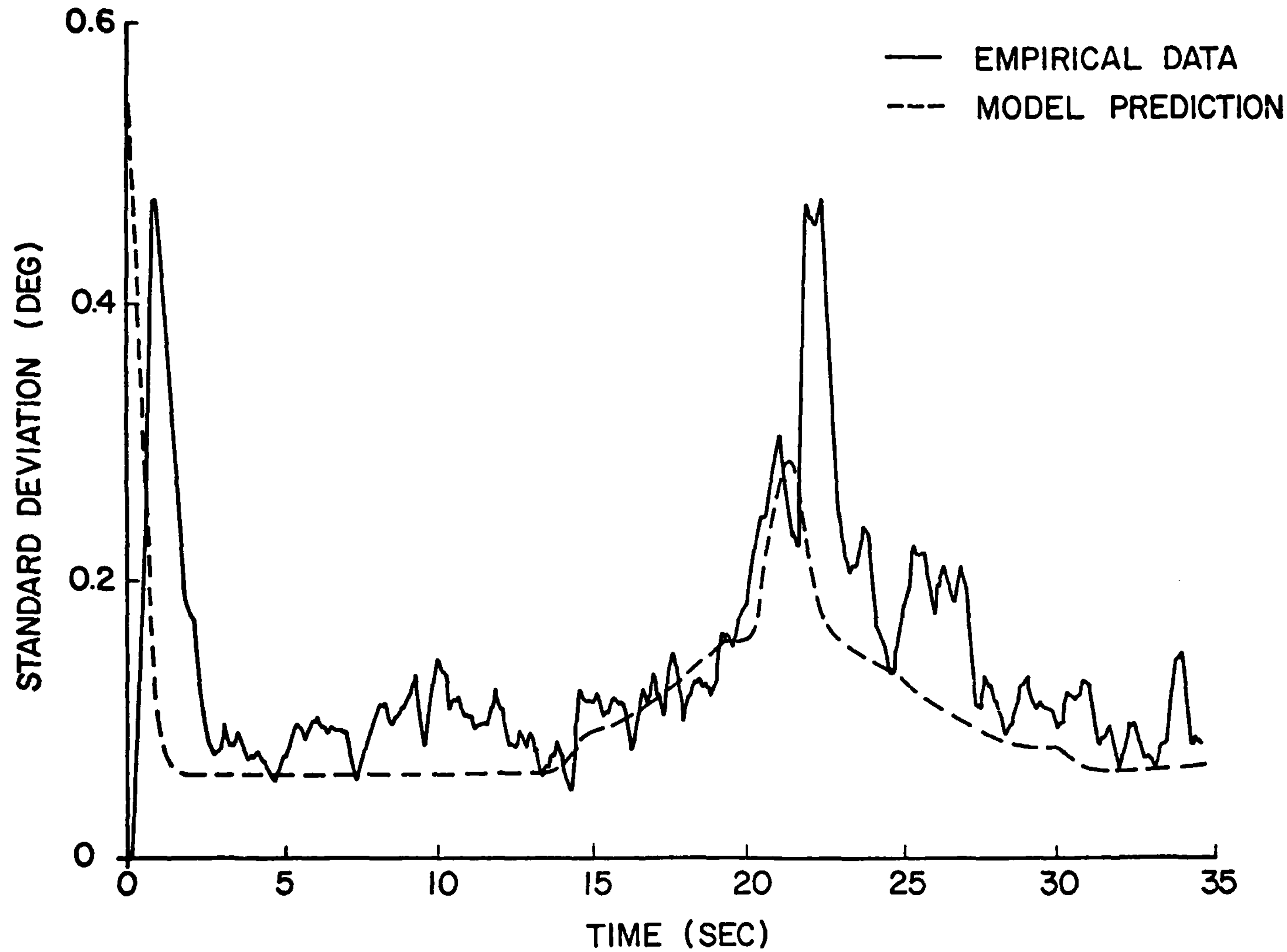


FIGURE 23 : STANDARD DEVIATION OF TRACKING ERROR — ELEVATION T5

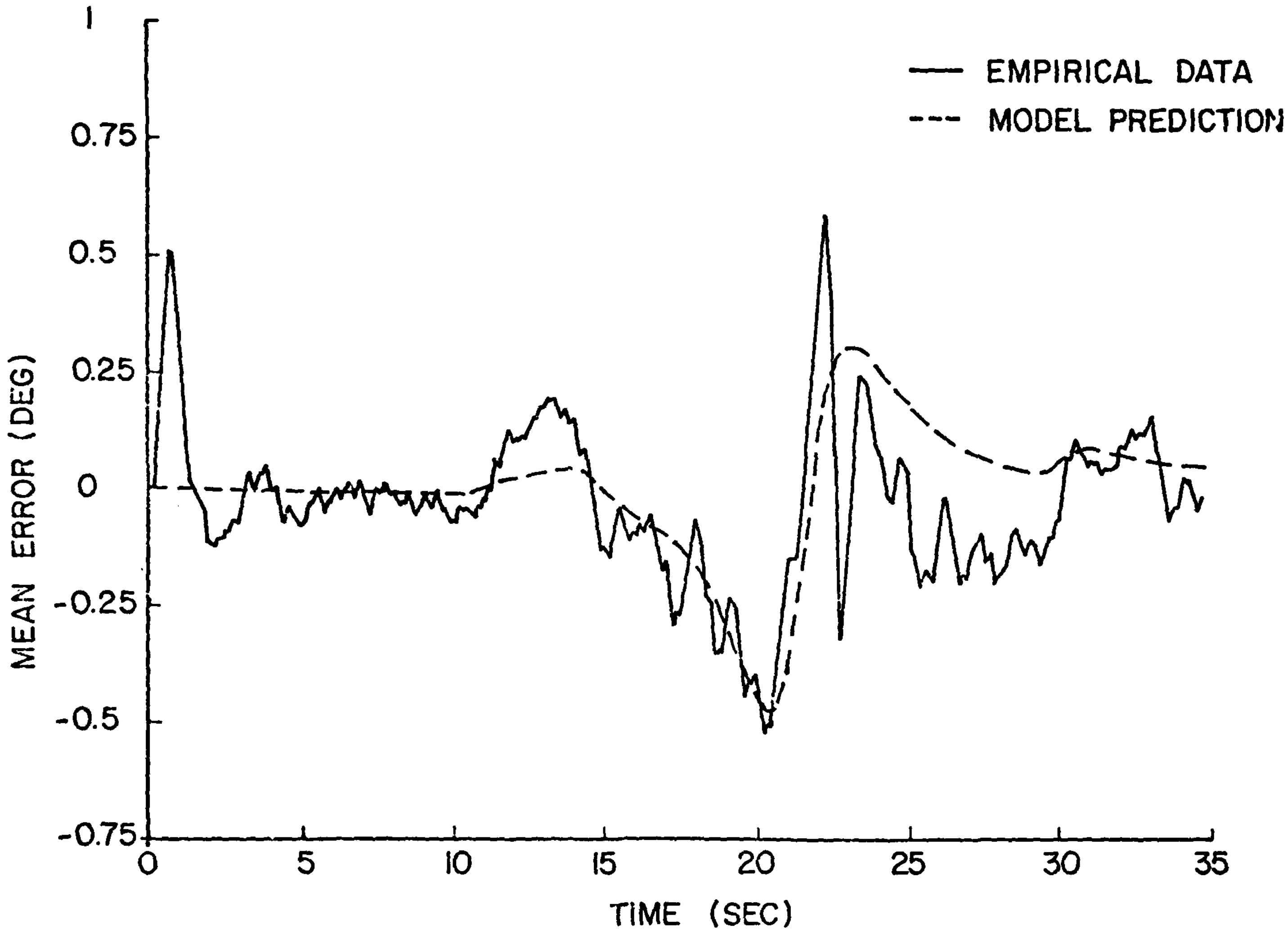


FIGURE 24: MEAN TRACKING ERROR
AZIMUTH T5

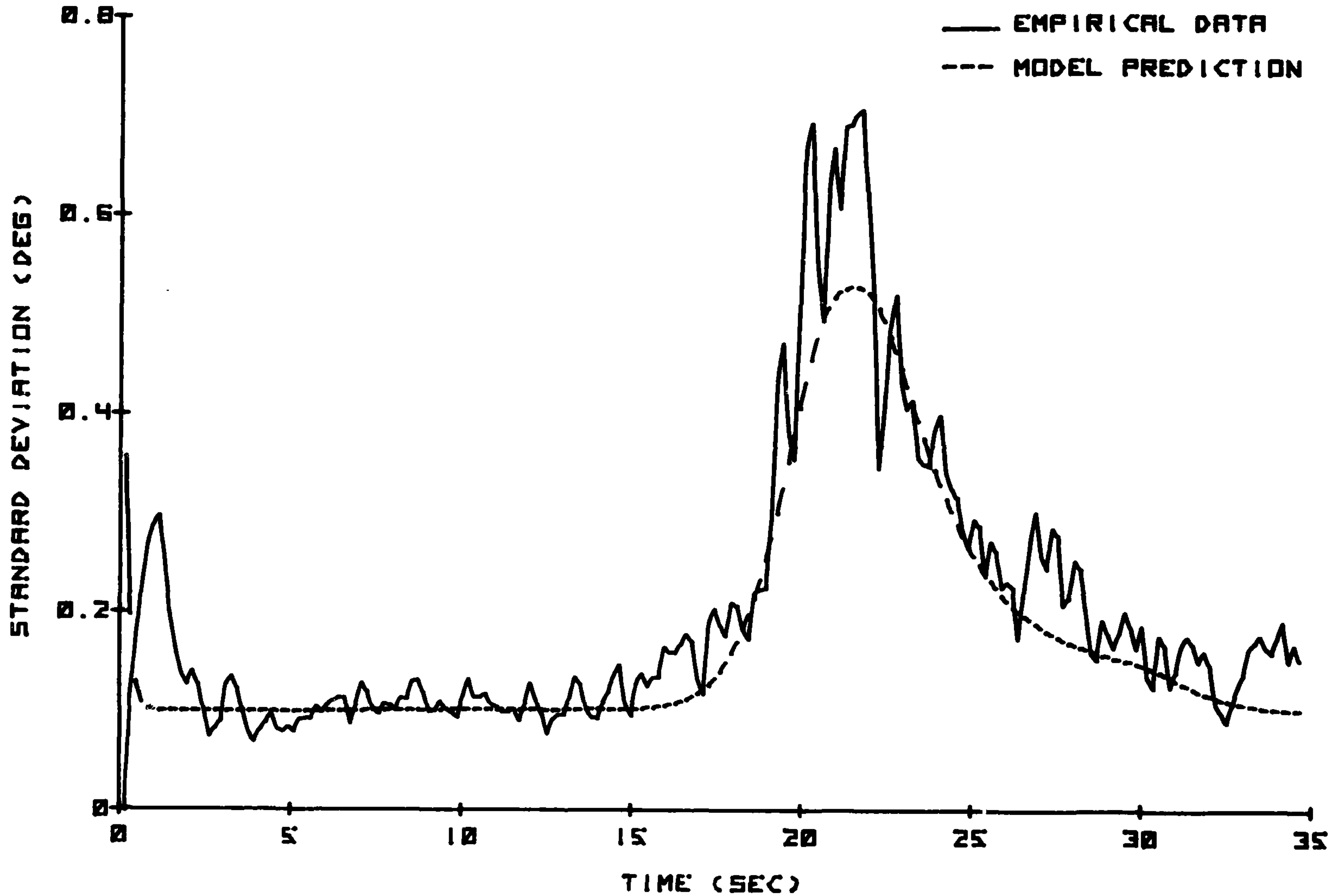


FIGURE 25: STANDARD DEVIATION OF TRACKING ERROR - AZIMUTH T5

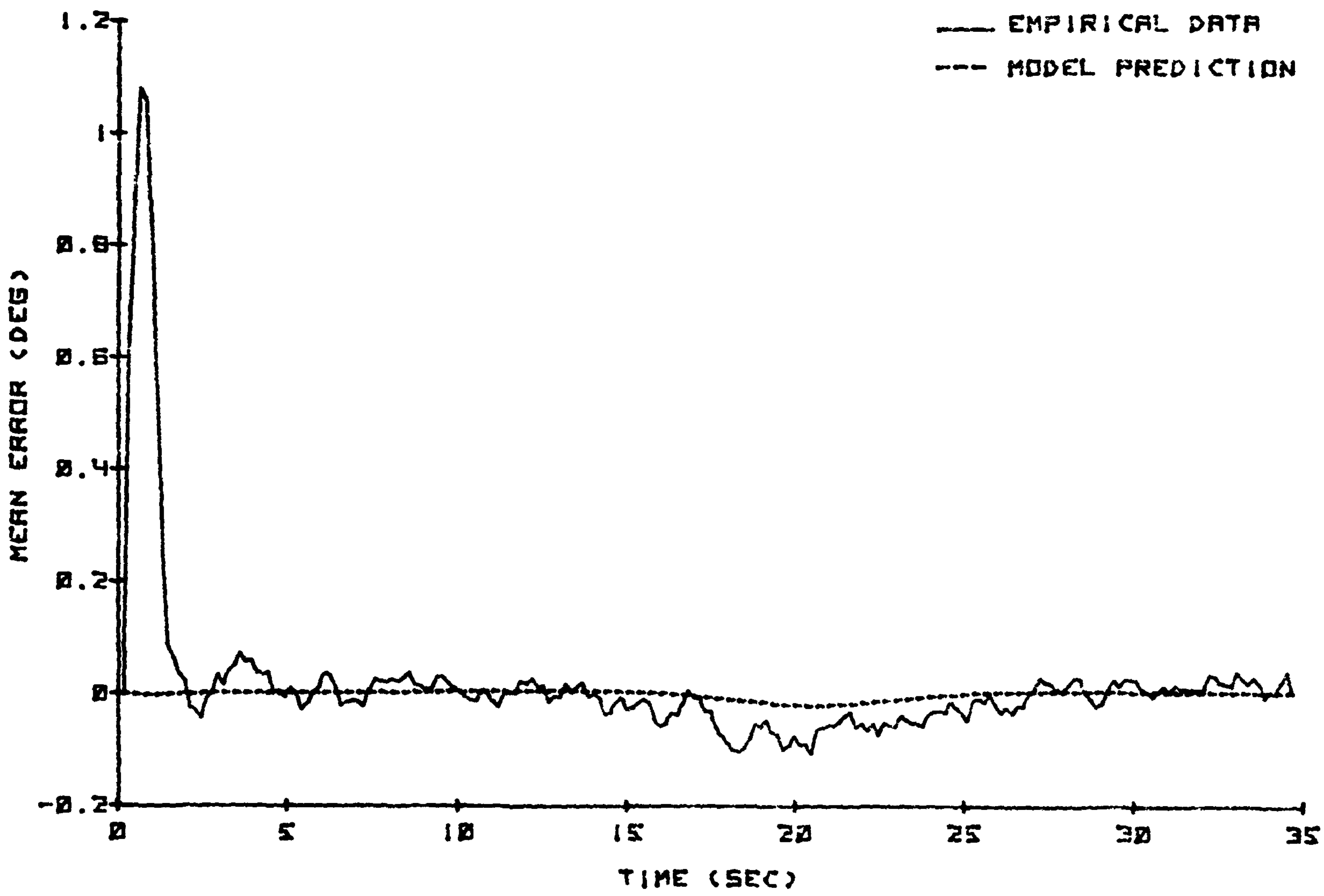


FIGURE 26: MEAN TRACKING ERROR - ELEVATION T6

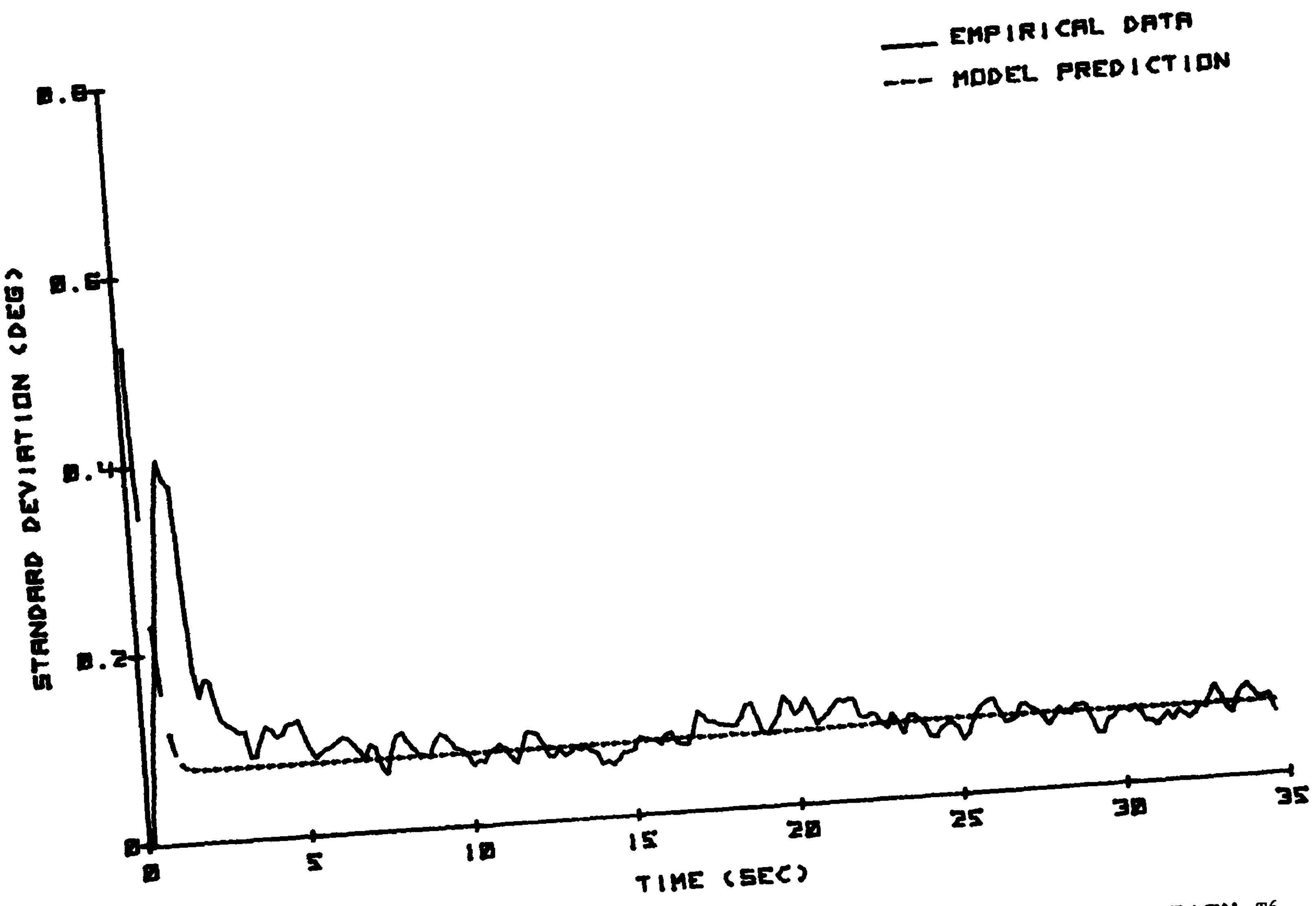


FIGURE 27 : STANDARD DEVIATION OF TRACKING ERROR - ELEVATION T6

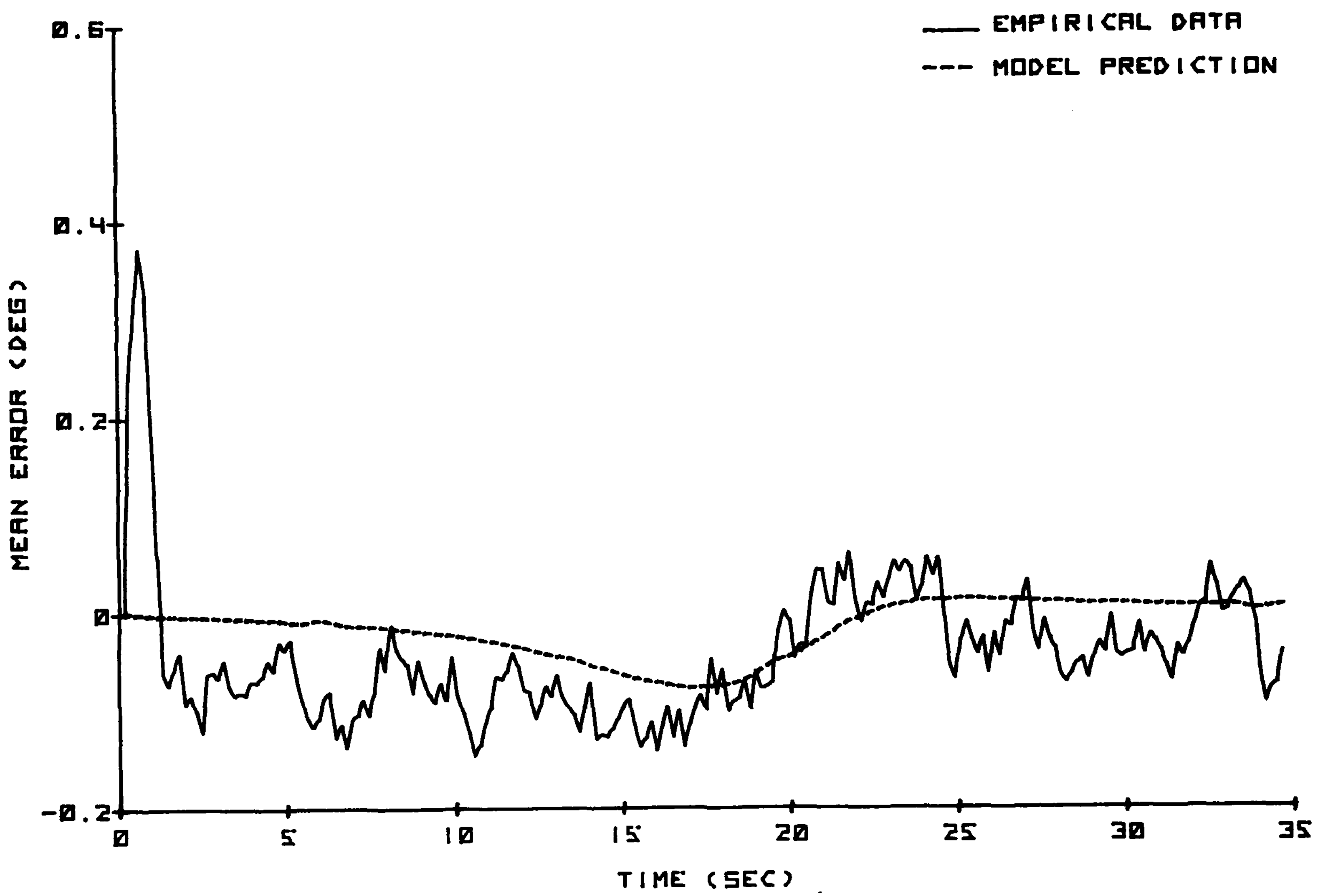


FIGURE 28: MEAN TRACKING ERROR - AZIMUTH T6

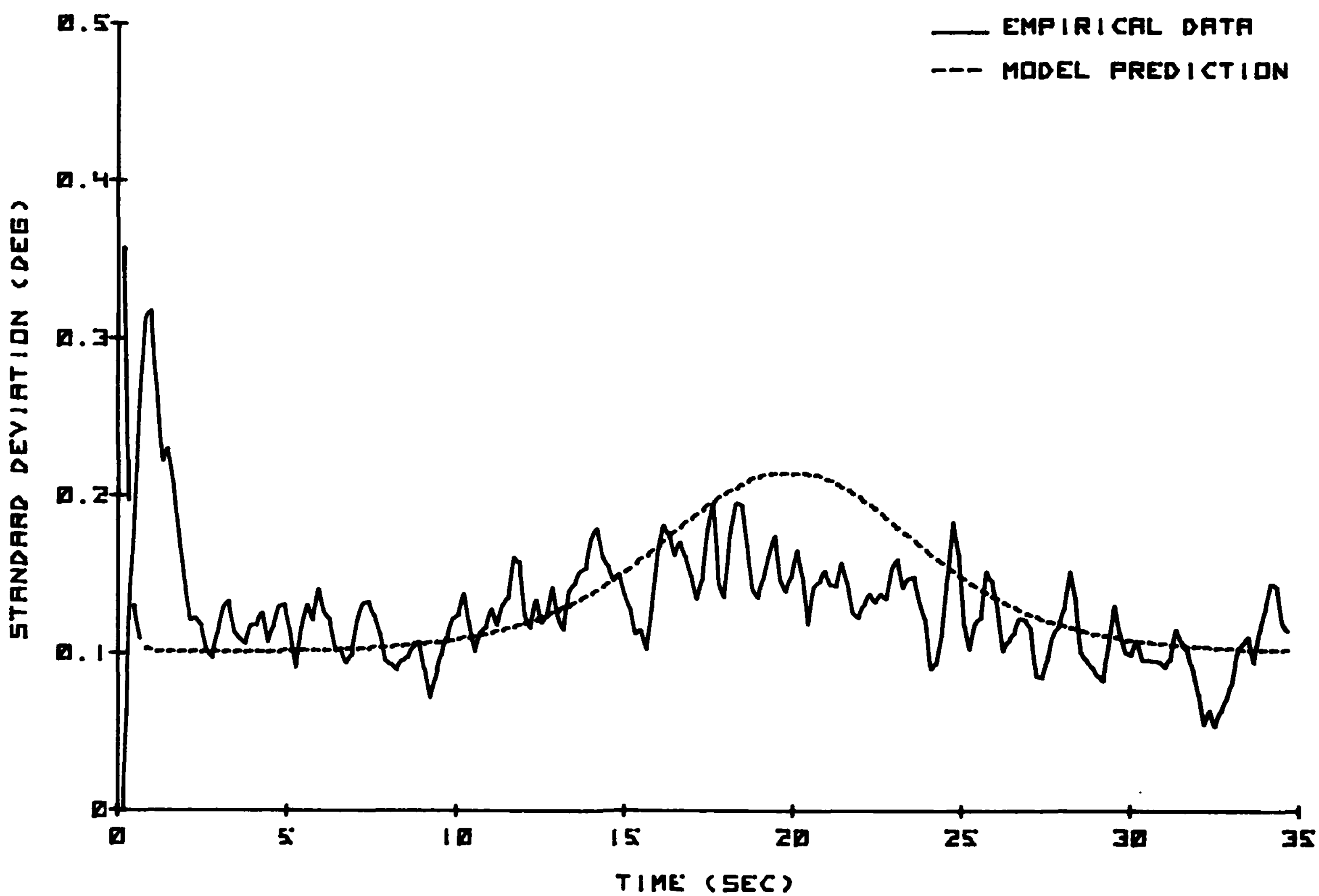


FIGURE 29 : STANDARD DEVIATION OF TRACKING ERROR - AZIMUTH T6

Section VI

CONCLUSION

This report presents some extended work on the development of an antiaircraft gunner model based on the observer theory for the MTQ system. In [2] through [8], the authors developed a gunner model for the study of weapon effectiveness of linear time-invariant AAA systems. The more general antiaircraft gunner model developed in this report can be as well applied to linear time-varying AAA systems (e.g., MTQ system). The structure of the gunner model contains three elements - a reduced-order ovserver, a feedback controller, and a remnant element. Here the reduced-order observer is designed as a linear time-varying system. A computer simulation program is developed with the designed model describing the gunner tracking performance for a linear time-varying AAA tracking system. Computer simulation results are in excellent agreement with the manned AAA simulation empirical data for several flyby and maneuvering trajectories. It is also shown that the gunner model can predict tracking error as accurately as the optimal control model. However, the computer execution time of the MTQ closed loop tracking system simulation utilizing the gunner model is much shorter than that using the optimal control model. All these results verify that the designed gunner model is an accurate and efficient model describing the gunner's compensatory tracking characteristics for various linear time-varying or time invariant AAA weapon systems.

The six model parameters--an observer gain, two controller gains, and three coefficients of the remnant covariance function--are determined by the combined least squares curve-fitting identification program. This identification program provides an efficient model validation method. The designed gunner model and the parameter identification procedure have been successfully applied to study the problems of the aircraft survivability and air defense weapon effectiveness at the Aerospace Medical Research Laboratory, Wright-Patterson AFB, Ohio. All the results of the computer simulation of closed-loop AAA tracking systems show that model parameters can be determined accurately by the curve-fitting program and the gunner model is a reliable representation of actual gunner's tracking performance.

APPENDIX A

Derivation of the Iterative Algorithm of Modified Gauss Newton Method:

The criterion function $J(\underline{a})$ of Equation (21) is rewritten here,

$$J(\underline{a}) = \int_0^{t_f} \left[\left(x_1'(t) - x_1'(t,\underline{a})\right)^2 + c\left(\overline{s}'(t) - \overline{s}(t,a)\right)^2\right] dt$$

Taking first order Taylor series expansions of $\overline{x}_1$ and $\overline{s}$ with respect to a certain initial guess $\underline{a}_o$, we have

$$J(\underline{a}) \cong \int_0^{t_f} \left\{\left[\overline{x}_1'(t) - \overline{x}_1\left(t,\underline{a}_o\right) - \frac{\partial \overline{x}_1\left(t,\underline{a}_o\right)}{\partial \underline{a}} \cdot \left(\underline{a} - \underline{a}_o\right)\right]^2 + \right.$$

$$\left. c\left[\overline{s}'(t) - \overline{s}\left(t,\underline{a}_o\right) - \frac{\partial \overline{s}\left(t,a_o\right)}{\partial \underline{a}} \cdot \left(\underline{a} - \underline{a}_o\right)\right]^2 \right\} dt$$

Next, the partial derivative of J with respect to $\underline{a}$ can be found as

$$\frac{\partial J(\underline{a})}{\partial \underline{a}} \cong \int_0^{t_f} \left\{-2\left(\overline{x}_1(t) - \overline{x}_1\left(t,\underline{a}_o\right)\right) \frac{\partial \overline{x}_1\left(t,a_o\right)}{\partial \underline{a}} + \right.$$

$$2\,\frac{\partial \overline{x}_1\left(t,a_o\right)}{\partial \underline{a}} \left(\underline{a} - \underline{a}_o\right) \cdot \frac{\partial \overline{x}_1^{\,T}\left(t,\underline{a}_o\right)}{\partial a} +$$

$$c\left[-2\left(\bar{s}'(t) - \bar{s}\left(t,\underline{a}_o\right)\right) \frac{\partial \bar{s}\ t,\underline{a}_o}{\partial \underline{a}} +\right.$$

$$\left.\left. 2\,\frac{\partial s\left(t,\underline{a}_o\right)}{\partial \underline{a}} \left(a - \underline{a}_o\right) \frac{\partial \bar{s}^T\left(t,\underline{a}_o\right)}{\partial \underline{a}} \right]\right\} dt \tag{A1}$$

Assume that $\underline{a}^*$ is a minimal, then

$$\frac{\partial J(\underline{a}^*)}{\partial \underline{a}} = 0 \tag{A2}$$

After discretization, Equations (A1) and (A2) can be rearranged to be

$$\underline{a}^* = \underline{a}_o - \left[\sum_{k=1}^{K} 2\left(\frac{\partial \bar{x}_1}{\partial \underline{a}}\right)^T \left(\frac{\partial \bar{x}_1}{\partial \underline{a}}\right) + c2\left(\frac{\partial \bar{s}}{\partial \underline{a}}\right)^T \left(\frac{\partial \bar{s}}{\partial \underline{a}}\right)\right]^{-1}$$

$$\cdot \sum_{k=1}^{K} \left\{-2\left(\bar{x}_1'(t) - \bar{x}_1\right)\left(\frac{\partial \bar{x}_1}{\partial \underline{a}}\right)^T\right.$$

$$\left. + c\left[-2\left(\bar{s}' - \bar{s}\right)\left(\frac{\partial \bar{s}}{\partial \underline{a}}\right)^T\right]\right\} \tag{A3}$$

Equation (A3) can be extended to a more general form,

$$\underline{a}_{i+1} = \underline{a}_i - \left[\sum_{k=1}^{K} 2\left(\frac{\partial x_1(t_k,\underline{a}_i)}{\partial \underline{a}}\right)^T \left(\frac{\partial \bar{x}_1(t_k,\underline{a}_i)}{\partial \underline{a}}\right) + c2 \cdot \left(\frac{\partial \bar{s}(t_k,\underline{a}_i)}{\partial \underline{a}}\right)^T \left(\frac{\partial \bar{s}(t_k,a_i)}{\partial \underline{a}}\right)\right]^{-1} \cdot$$

$$\sum_{k=1}^{K} \left\{-2\left(\bar{x}_1'(t_k) - \bar{x}_1(t_k,\underline{a}_i)\right)\left(\frac{\partial \bar{x}_i \; t_k,\underline{a}_i}{\partial \underline{a}}\right)^T + c\left[-2\left(\bar{s}'(t_k) - \bar{s}(t_k,\underline{a}_i)\right)\left(\frac{\partial s \; t_k,\underline{a}_i}{\partial \underline{a}}\right)^T\right]\right\}$$

$i = 0, 1, 2, 3, \ldots\ldots$

This is the modified Gauss Newton iterative gradiant algorithm.

APPENDIX B

PRGRAM LISTING OF PARAMETER IDENTIFICATION PROCEDURES-AZIMUTH CASE

```
WBM,T600,IO100,CM110000,STC58. L7-6230,MORAN,256-3560
FTN.
COMMENT. **** NO DECK ****
ATTACH,TAPE1,DAT10,CY=2.
ATTACH,IMSL,ID=LIBRARY,SN=ASD,MR=1.
LIBRARY,IMSL.
LGO.
      PROGRAM AZFIT(INPUT,OUTPUT,TAPE1)
      COMMON/VAR/DT,N1,N2,IP,LIM1,LIM2,EE,WF
      COMMON/MAT1/F2(512,5),A(6),AA(6),B(512,14)
      COMMON/MAT2/F3(2048),D(6),PHID(512)
      DATA IP,WF/6,1.0/
C USE EVERY "I1" DATA POINT
      READ*,I1
      N1=90/I1
      N2=1+1049/I1
      IF(N2.GT.512) N2=512
      DT=.033*I1
      PRINT*,3HI1=,I1,3HDT=,DT,3HN1=,N1,3HN2=,N2
      DO 15 I=1,2
      IC1=0
      DO 10 KI=1,1050
      Z60=Z6
      READ(1,*)Z1,Z2,Z3,Z4,Z5,Z6
      KK=KI+I1-1
      IF(MOD(KK,I1).NE.0) GO TO 10
      IC1=IC1+1
      IF(IC1.GT.512) GO TO 10
      IF(I.EQ.2) GO TO 5
      F2(IC1,1)=Z1
      F2(IC1,2)=Z2
      F2(IC1,3)=Z6
      B(IC1,1)=Z3
      PHID(IC1)=(Z6-Z60)/.033
      GO TO 10
    5 Z2=Z2/(1000.*COS(B(IC1,1)))
      Z4=Z4/(1000.*COS(B(IC1,1)))
      F2(IC1,4)=Z2
      F2(IC1,5)=Z4
   10 CONTINUE
   15 CONTINUE
C INPUT "1" TO GO ON "0" TO STOP
   20 READ*,IFG
      IF(IFG.EQ.0) STOP
C MAKE INITIAL GUESS AND INPUT ITERATION LIMITS
C INPUT LIM1, LIM2, & EE
      READ*,LIM1,LIM2,EE
      PRINT*,5HLIM1=,LIM1,5HLIM2=,LIM2,3HEE=,EE
C INPUT 1ST GUESS AS KGAIN, GAMMA1, GAMMA2, ALPHA1, ALPHA2, ALPHA3
      READ*,AA
      PRINT*,9H1ST GUESS,AA
C COMPUTE INITIAL COST FUNCTION
```

```
      CALL COEF(AA,RJ2)
      PRINT*,6HJMIN1=,RJ2
      CALL LOOP(RJ2)
      GO TO 20
      END
      SUBROUTINE LOOP(RJ2)
C ITERATION PROCESS
      DIMENSION R(6),Q(6,6),QI(6,6),WORK(6),DDD(6)
      COMMON/VAR/DT,N1,N2,IP,LIM1,LIM2,EE,WF
      COMMON/MAT1/F2(512,5),A(6),AA(6),B(512,14)
      COMMON/MAT2/F3(2048),D(6),PHID(512)
      NCT=MCT=0 $ DEL=1.0
    5 DO 10 I=1,IP
      R(I)=0.
      D(I)=0.
      DO 10 J=1,IP
   10 Q(I,J)=0.
      DO 20 K=N1,N2
      EDIF=-2.*(F2(K,4)-B(K,7))
      SDIF=-2.*WF*(F2(K,5)-B(K,14))
      DO 15 I=1,IP
      R(I)=R(I)+B(K,I)*EDIF+B(K,I+7)*SDIF
      DO 15 J=1,IP
   15 Q(I,J)=Q(I,J)+2.*B(K,I)*B(K,J)+2.*WF*B(K,I+7)*B(K,J+7)
   20 CONTINUE
C USE IMSL LIB ROUTINE "LINV1F" TO COMPUTE INVERSE OF Q
      CALL LINV1F(Q,IP,IP,QI,0,WORK,IER)
      IF(IER.EQ.129) PRINT*,4HIER=,IER
C COMPUTE D MATRIX FROM Q INVERSE AND R
      DO 25 I=1,IP
      DO 25 J=1,IP
   25 D(I)=-QI(I,J)*R(J)+D(I)
C COMPUTE A(I+1)=A(I)+D(I)
   30 DO 32 I=1,IP
   32 A(I)=AA(I)+D(I)
      J=0
      IF(A(1).LT.0.) J=1
      IF(A(2).GT.0.) J=2
      IF(A(3).GT.0.) J=3
      IF(A(4).LT.0.) J=4
      IF(A(5).LT.0.) J=5
      IF(A(6).LT.0.) J=6
      IF(J.NE.0) GO TO 34
      CALL COEF(A,RJ1)
      IF(RJ1.LE.RJ2) GO TO 40
   34 DEL=.5
      DO 36 I=1,IP
   36 D(I)=D(I)*DEL
   38 NCT=NCT+1
      IF(NCT.LE.LIM1) GO TO 30
      PRINT*,7HERROR 1,10H      NCT=,NCT,5H MCT=,MCT
      PRINT*,3H A=,A
      PRINT*,5H RJ1=,RJ1,5H RJ2=,RJ2
```

```
      RETURN
C DETERMINE IF A(I+1) IS ACCEPTABLE
   40 DO 45 I=1,IP
      DDD(I)=ABS(D(I)/AA(I))
   45 IF(DDD(I).GT.EE) GO TO 50
      GO TO 65
   50 MCT=MCT+1
      IF(MCT.LE.LIM2) GO TO 55
      PRINT*,7HERROR 2,10H      NCT=,NCT,5H MCT=,MCT
      PRINT*,3H A=,A
      PRINT*,5H RJ1=,RJ1,5H RJ2=,RJ2
      RETURN
   55 RJ2=RJ1
      NCT=0
      DEL=1.
      DO 60 I=1,IP
   60 AA(I)=A(I)
      GO TO 5
   65 PRINT 70,A,NCT,MCT
   70 FORMAT(1X,6G12.5,2X,2I5)
      PRINT*,3HAA=,AA
      PRINT*,3H D=,D
      PRINT*,5HJMIN=,RJ1
      END
      SUBROUTINE COEF(W,RJ)
      DIMENSION W(6),IWK(11),BW(1024)
      DIMENSION THD(512),THDD(512),THDK(512),THDDK(512)
      COMMON/VAR/DT,N1,N2,IP,LIM1,LIM2,EE,WF
      COMMON/MAT1/F2(512,5),A(6),AA(6),B(512,14)
      COMMON/MAT2/F3(2048),D(6),PHIL(512)
      REAL K
      K=W(1) $ GAM1=W(2) $ GAM2=W(3)
      ALF1=W(4) $ ALF2=W(5) $ ALF3=W(6)
      DO 10 I=1,N2
      T=(I-1)*DT
      BW(I)=F2(I,2)
      C1=COS(F2(I,3))
      ARG=C1*K*T
      S1=0.
      IF(ARG.LT.200.) S1=EXP(-ARG)
   10 F3(I)=S1
      CALL VCONVO(F3,BW,N2,N2,IWK)
      DO 20 I=1,N2
      THD(I)=F2(I,1)-F3(I)*DT
      IF(I.EQ.1) GO TO 20
      THDD(I)=(THD(I)-THD(I-1))/DT
   20 CONTINUE
      THDD(1)=THDD(2)
      DO 30 I=1,N2
      T=(I-1)*DT
      BW(I)=F2(I,2)
      C1=COS(F2(I,3))
      C2=-TAN(F2(I,3))*PHIL(I)
```

```
      X1=C2+1.28*C1*GAM1
      X2=X1+K*C1
      ARG1=C1*K*T
      ARG2=X1*T
      S1=0. $ S2=0.
      IF(ARG1.LT.200.) S1=EXP(-ARG1)
      IF(ARG2.GT.-200.) S2=EXP(ARG2)
      X3=-T*X2-1.
   30 F3(I)=(1.28*C1*C1*GAM2)*(S1*X3+S2)/X2**2
      CALL VCONVO(F3,BW,N2,N2,IWK)
      DO 40 I=1,N2
      T=(I-1)*DT
      BW(I)=F2(I,2)
      B(I,1)=F3(I)*DT
      C1=COS(F2(I,3))
      C2=-TAN(F2(I,3))*PHID(I)
      X1=C2+1.28*C1*GAM1
      X2=X1+K*C1
      X3=1.28*C1*(C1+1.28*C1*GAM2)/X1**2
      X4=(1.28*C1)**2*GAM2/X2**2
      ARG1=C1*K*T
      ARG2=X1*T
      S1=0. $ S2=0.
      IF(ARG1.LT.200.) S1=EXP(-ARG1)
      IF(ARG2.GT.-200.) S2=EXP(ARG2)
   40 F3(I)=X3*(S2*(X1*T-1.)+1.)+X4*(S2*(-X2*T+1.)-S1)
      CALL VCONVO(F3,BW,N2,N2,IWK)
      DO 50 I=1,N2
      T=(I-1)*DT
      BW(I)=F2(I,2)
      B(I,2)=F3(I)*DT
      C1=COS(F2(I,3))
      C2=-TAN(F2(I,3))*PHID(I)
      X1=C2+1.28*C1*GAM1
      X2=X1+K*C1
      X3=1.28*C1/X1
      X4=1.28*C1/X2
      ARG1=C1*K*T
      ARG2=X1*T
      S1=0. $ S2=0.
      IF(ARG1.LT.200.) S1=EXP(-ARG1)
      IF(ARG2.GT.-200.) S2=EXP(ARG2)
   50 F3(I)=X3*(S2-1.)-X4*(S2-S1)
      CALL VCONVO(F3,BW,N2,N2,IWK)
      DO 60 I=1,N2
      B(I,3)=F3(I)*DT
      B(I,4)=0.
      B(I,5)=0.
      B(I,6)=0.
      T=(I-1)*DT
      BW(I)=F2(I,2)
      C1=COS(F2(I,3))
      C2=-TAN(F2(I,3))*PHID(I)
```

```
      X1=C2+1.28*C1*GAM1
      X2=X1+K*C1
      X3=(C1+1.28*C1*GAM2)/X1
      X4=-1.28*C1*GAM2/X2
      ARG1=C1*K*T
      ARG2=X1*T
      S1=0. $ S2=0.
      IF(ARG1.LT.200.) S1=EXP(-ARG1)
      IF(ARG2.GT.-200.) S2=EXP(ARG2)
   60 F3(I)=X3*(S2-1.)+X4*(S2-S1)
      CALL VCONVO(F3,BW,N2,N2,IWK)
      DO 70 I=1,N2
      B(I,7)=F3(I)*DT
      T=(I-1)*DT
      BW(I)=F2(I,2)
      C1=COS(F2(I,3))
      ARG1=C1*K*T
      S1=0.
      IF(ARG1.LT.200.) S1=EXP(-ARG1)
   70 F3(I)=C1*T*S1
      CALL VCONVO(F3,BW,N2,N2,IWK)
      DO 80 I=1,N2
      THDK(I)=F3(I)*DT
      IF(I.EQ.1) GO TO 80
      THDDK(I)=(THDK(I)-THDK(I-1))/DT
   80 CONTINUE
      THDDK(1)=THDDK(2)
      DO 85 I=1,N2
      T=(I-1)*DT
      BW(I)=ALF1+ALF2*THD(I)**2+ALF3*THDD(I)**2
   85 F3(I)=FT(F2(I,3),PHID(I),T,GAM1,GAM2,K)
      CALL VCONVO(F3,BW,N2,N2,IWK)
      DO 90 I=1,N2
      B(I,14)=SQRT(F3(I)*DT)
      T=(I-1)*DT
      BW(I)=2.*ALF2*THD(I)*THDK(I)+2.*ALF3*THDD(I)*THDDK(I)
   90 F3(I)=FT(F2(I,3),PHID(I),T,GAM1,GAM2,K)
      CALL VCONVO(F3,BW,N2,N2,IWK)
      DO 100 I=1,N2
      T=(I-1)*DT
      B(I,8)=F3(I)*DT
      BW(I)=ALF1+ALF2*THD(I)**2+ALF3*THDD(I)**2
      C1=COS(F2(I,3))
      C2=-TAN(F2(I,3))*PHID(I)
      X1=C2+1.28*C1*GAM1
      X2=X1+K*C1
      S1=0. $ S2=0.
      ARG1=C1*K*T
      ARG2=X1*T
      IF(ARG1.LT.200.) S1=EXP(-ARG1)
      IF(ARG2.GT.-200.) S2=EXP(ARG2)
      X3=1.28*C1*K*GAM2/X2
      X4=(1.28*C1)**2*2.*((1.+X3)*S2-X3*S1)
```

```
      X5=(X2*1.28*C1*GAM2-1.28*C1**2*K*GAM2)/X2**2
      X6=(1.28*C1*C1*K*GAM2*T)/X2
  100 F3(I)=X4*(S2*X5+X6*S1-S1*X5)
      CALL VCONVO(F3,BW,N2,N2,IWK)
      DO 110 I=1,N2
      T=(I-1)*DT
      BW(I)=ALF1+ALF2*THD(I)**2+ALF3*THDD(I)**2
      B(I,8)=B(I,8)+F3(I)*DT
      C1=COS(F2(I,3))
      C2=-TAN(F2(I,3))*PHID(I)
      X1=C2+1.28*C1*GAM1
      X2=X1+K*C1
      S1=0. $ S2=0.
      ARG1=C1*K*T
      ARG2=X1*T
      IF(ARG1.LT.200.) S1=EXP(-ARG1)
      IF(ARG2.GT.-200.) S2=EXP(ARG2)
      X3=1.28*C1*K*GAM2/X2
      X4=1.28*C1*K*GAM2/X2**2
      X5=2.*((1.+X3)*S2-X3*S1)*(1.28*C1)**3
  110 F3(I)=X5*(S2*((1.+X3)*T-X4)+S1*X4)
      CALL VCONVO(F3,BW,N2,N2,IWK)
      DO 120 I=1,N2
      T=(I-1)*DT
      BW(I)=ALF1+ALF2*THD(I)**2+ALF3*THDD(I)**2
      B(I,9)=F3(I)*DT
      C1=COS(F2(I,3))
      C2=-TAN(F2(I,3))*PHID(I)
      X1=C2+1.28*C1*GAM1
      X2=X1+K*C1
      S1=0. $ S2=0.
      ARG1=C1*K*T
      ARG2=X1*T
      IF(ARG1.LT.200.) S1=EXP(-ARG1)
      IF(ARG2.GT.-200.) S2=EXP(ARG2)
      X3=1.28*C1*K/X2
      X4=X3*GAM2
      X5=(1.28*C1)**2*2.*((1.+X4)*S2-X4*S1)
  120 F3(I)=X5*(S2*X3-S1*X3)
      CALL VCONVO(F3,BW,N2,N2,IWK)
      DO 130 I=1,N2
      T=(I-1)*DT
      BW(I)=1.0
      B(I,10)=F3(I)*DT
  130 F3(I)=FT(F2(I,3),PHID(I),T,GAM1,GAM2,K)
      CALL VCONVO(F3,BW,N2,N2,IWK)
      DO 140 I=1,N2
      T=(I-1)*DT
      BW(I)=THD(I)**2
      B(I,11)=F3(I)*DT
  140 F3(I)=FT(F2(I,3),PHID(I),T,GAM1,GAM2,K)
      CALL VCONVO(F3,BW,N2,N2,IWK)
      DO 150 I=1,N2
```

```
      T=(I-1)*DT
      BW(I)=THDD(I)**2
      B(I,12)=F3(I)*DT
  150 F3(I)=FT(F2(I,3),PHID(I),T,GAM1,GAM2,K)
      CALL VCONVO(F3,BW,N2,N2,IWK)
      DO 160 I=1,N2
  160 B(I,13)=F3(I)*DT
      DO 165 J=8,13
      DO 165 I=1,N2
  165 B(I,J)=.5*B(I,J)/B(I,14)
      RJ=0.
      DO 170 I=N1,N2
  170 RJ=RJ+(F2(I,4)-B(I,7))**2+WF*(F2(I,5)-B(I,14))**2
      END
      FUNCTION FT(THT,PHD,T,GAM1,GAM2,K)
      C1=COS(THT)
      C2=-TAN(THT)*PHD
      X1=C2+1.28*C1*GAM1
      X2=X1+K*C1
      S1=0. $ S2=0.
      ARG1=C1*K*T
      ARG2=X1*T
      TF(ARG1.LT.200.) S1=EXP(-ARG1)
      IF(ARG2.GT.-200.) S2=EXP(ARG2)
      X3=1.28*C1*GAM2*K/X2
      FT=(1.28*C1)**2*((1.+X3)*S2-X3*S1)**2
      END
2
1
30,50,.05
1,-5,-1,.1,.01,.001
0
```

APPENDIX C

PROGRAM LISTING OF PARAMETER IDENTIFICATION PROCEDURES-ELEVATION CASE

```
     HELP,CY=3,ID=L740238
W8F,T800,IO400,CM70000,STCSB. L740238,GLASS,25#-3960
FTN,L=0.
ATTACH,IMSL,ID=LIBRARY,SN=ASD.
LIBRARY,IMSL.
ATTACH,T,DAT10,CY=1,MR=1.
COPY,T,TAPE1.
LGO.
      PROGRAM FIF(INPUT,OUTPUT,TAPE1)
      COMMON/VAR/LIM1,LIM2,EE,IP,H,N,DEL,N1,CE
      COMMON/MAT/F2(512,4),A(6),AA(6),B(512,9),D(6),F3(512,4)
      REWIND 1
C     CURVE FIT PROGRAM FOR ROOMS - MTQ - ELEVATION
C     PARAMETERS IDENTIFIED ARE:
C          A(1)=KGAIN     A(2)=GAMMA1     A(3)=GAMMA2
C          A(4)=ALPHA1    A(5)=ALPHA2     A(6)=ALPHA3
C     SAMPLES OF DATA ARE SELECTED AT EVERY I1 POINTS
C     TIME STEP = I1*.033
C     FIRST THREE SECONDS OF DATA ARE CONSIDERED AQUISITION AND NOT USE
C     CE=STANDARD DEVIATION WEIGHTING FACTOR  (CE=1.)
      PRINT*,30H PICK EVERY -- SAMPLES
      READ*,I1
      N=1+1049/I1
      IF(N.GT.512)N=512
      H=.033*I1
      N1=90/I1
      IP=6
      PRINT*,3HI1=,I1,4HDEL=,H,3H N=,N,3HN1=,N1,4H IP=,IP
C     READ IN TRAJECTORY INFORMATION AND EMPIRICAL DATA
C     ALL DATA IN RADIANS
      DO 6 I=1,2
      IC1=0
      DO 6 KI=1,1050
2     READ(1,*)Z1,Z2,Z3,Z4,Z5,Z6
      KK=KI+I1-1
      IF(MOD(KK,I1).NE.0)GO TO 6
4     IC1=IC1+1
      IF(IC1.GT.512)GO TO 6
      IF(I.EQ.1)GO TO 5
      Z4=Z3/1000.
      Z5=Z5/1000.
5     II=(I-1)*2+1
      F2(IC1,II)=Z4
      F2(IC1,II+1)=Z5
6     CONTINUE
11    PRINT*,50HTYPE 1 TO GO OR TYPE 0 TO STOP
      READ*,IFG
      IF(IFG.EQ.0)STOP
      CALL INIT
      GO TO 11
      END
      SUBROUTINE INIT
```

```
C       INITIALIZATION ROUTINE
C       INPUT OVERRIDE LIMITS AND EPSILON  (EE=.01)
C       MAKE INITIAL GUESS
        COMMON/VAR/LIM1,LIM2,EE,IP,H,N,DEL,N1,CE
        COMMON/MAT/F2(512,4),A(6),AA(6),B(512,9),D(6),F3(512,4)
        PRINT*,30H TYPE IN LIM1,LIM2,EE
        READ*,LIM1,LIM2,EE
        CE=1.
        PRINT*,5HLIM1=,LIM1,5HLIM2=,LIM2,3HEE=,EE,3HCE=,CE
        PRINT*,25HTYPE IN K,G1,G2,A1,A2,A3
        READ*,AA
        PRINT*,10H 1ST GUESS,AA
        DEL=1
        CALL COEF(AA,RJ2)
        PRINT*,7H JMIN1=,RJ2
        CALL LOOP(RJ2)
        END
        SUBROUTINE DD
C       CALCULATE D MATRIX FROM Q AND R
        DIMENSION R(6),Q(6,6),W1(6,6)
        COMMON/VAR/LIM1,LIM2,EE,IP,H,N,DEL,N1,CE
        COMMON/MAT/F2(512,4),A(6),AA(6),B(512,9),D(6),F3(512,4)
        COMMON/MAIN1/NDIM,NDIM1
        NDIM=IP$NDIM1=IP+1
        DO 10 I=1,IP
        R(I)=0.
        DO 10 J=1,IP
10      Q(I,J)=0.
        DO 35 K=N1,N
        SN=F2(K,3)-F3(K)
        SN1=F2(K,4)-F3(K,2)
        DO 30 I=1,IP
        DO 25 J=1,IP
        Q(I,J)=Q(I,J)+2.*CE*B(K,I)*B(K,J)
        IF(I.GT.3.OR.J.GT.3)GO TO 25
        Q(I,J)=Q(I,J)+2.*B(K,I+6)*B(K,J+6)
25      CONTINUE
        R(I)=R(I)-2.*CE*SN1*B(K,I)
        IF(I.GT.3)GO TO 30
        R(I)=R(I)-2.*SN*B(K,I+6)
30      CONTINUE
35      CONTINUE
        CALL GMINV(IP,IP,Q,W1,MR,1)
        IP1=IP*IP
        DO 50 I=1,IP
        II=1
        D(I)=0.
        DO 45 J=I,IP1,IP
        D(I)=D(I)+W1(J)*R(II)
45      II=II+1
50      D(I)=-D(I)
        END
        SUBROUTINE LOOP(RJ2)
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDQ

```
C     ITERATION PROCESS
C     COMPUTE A(I+1)=A(I)+D(I)
C     DETERMINE WHEN A(I+1) IS ACCEPTABLE
      DIMENSION DDD(6)
      COMMON/VAR/LIM1,LIM2,EE,IP,H,N,DEL,N1,CE
      COMMON/MAT/F2(512,4),A(6),AA(6),B(512,9),D(6)
      NCT=MCT=0
1     CALL DD
2     DO 100 I=1,IP
      D(I)=D(I)*DEL
100   A(I)=AA(I)+D(I)
      IF(A(1).LT.0..OR.A(2).GT.0.)GO TO 15
      IF(A(4).LT.0..OR.A(5).LT.0..OR.A(6).LT.0.)GO TO 15
      CALL COEF(A,RJ1)
6     IF(RJ1.LT.RJ2)GO TO 30
15    DEL=.5
      NCT=NCT+1
      IF(NCT.LE.LIM1)GO TO 2
      PRINT*,7HERROR 1,3H A=,A,5H NCT=,NCT,5H MCT=,MCT,RJ1,RJ2
      PRINT*,3H D=,D,3HAA=,AA
      RETURN
30    DO 35 I=1,IP
      DDD(I)=ABS(D(I)/AA(I))
35    IF(DDD(1).GT.EE)GO TO 20
      GO TO 40
20    MCT=MCT+1
      IF(MCT.LE.LIM2)GO TO 9
      PRINT*,7HERROR 2,3H A=,A,5H NCT=,NCT,5H MCT=,MCT,RJ1,RJ2
      PRINT*,3H D=,D,3HAA=,AA
      RETURN
9     RJ2=RJ1
      NCT=0
      DEL=1.
      DO 25 I=1,IP
25    AA(I)=A(I)
      GO TO 1
40    PRINT45,(A(I),I=1,IP),NCT,MCT
      PRINT*,3HAA=,AA
      PRINT*,3H D=,D
      PRINT*,5HJMIN=,RJ1
45    FORMAT(1X,6G12.5,2X,2I5)
      END
      SUBROUTINE COEF(W,RJ)
      DIMENSION W(6),IWK(11)
      COMMON/VAR/LIM1,LIM2,EE,IP,H,N,DEL,N1,CE
      COMMON/MAT/F2(512,4),A(6),AA(6),B(512,9),D(6),F3(512,4)
      U=W(1)
      V=W(2)
      Y=W(3)
      A1=W(4)
      A2=W(5)
      A3=W(6)
      C=1.34
```

```
      X=C*V+U
      X1=1+C*Y
      REWIND 1
C     COMPUTE PARTIAL DERIVATIVE OF MEAN ERROR WRT KGAIN
      DO 10 I=1,N
      T=(I-1)*H
      B(I)=F2(I,2)
      S1=S2=100.
      IF(U*T.GE.200.)S1=0.
      IF(C*V*T.LE.-200.)S2=0.
      IF(S1.NE.0.)S1=EXP(-U*T)
      IF(S2.NE.0.)S2=EXP(C*V*T)
      F3(I)=(S2*X1-X1-C*C*V*Y*T*S1)/(X*C*V)
10    F3(I)=F3(I)-((U*X1+C*V)*S2-X*X1+C*C*V*Y*S1)/(C*V*X*X)
      CALL VCONVO(F3,B,N,N,IWK)
C     COMPUTE PARTIAL DERIVATIVE OF MEAN ERROR WRT GAMMA1
      DO 15 I=1,N
      FF=F3(I)*H
      WRITE(1)FF
      B(I)=F2(I,2)
      T=(I-1)*H
      S1=S2=100.
      IF(U*T.GE.200.)S1=0.
      IF(C*V*T.LE.-200.)S2=0.
      IF(S1.NE.0.)S1=EXP(-U*T)
      IF(S2.NE.0.)S2=EXP(C*V*T)
      F3(I)=C*V*X*(S2*(1+T*(X+C*U*Y))-X1+C*Y*S1)
      F3(I)=F3(I)-(U+2.*C*V)*((U*X1+C*V)*S2-X*X1+C*C*V*Y*S1)
15    F3(I)=F3(I)/(C*V*V*X*X)
      CALL VCONVO(F3,B,N,N,IWK)
C     COMPUTE PARTIAL DERIVATIVE OF MEAN ERROR WRT GAMMA1
      DO 17 I=1,N
      FF=F3(I)*H
      WRITE(1)FF
      B(I)=F2(I,2)
      T=(I-1)*H
      S1=S2=100.
      IF(U*T.GE.200.)S1=0.
      IF(C*V*T.LE.-200.)S2=0.
      IF(S1.NE.0.)S1=EXP(-U*T)
      IF(S2.NE.0.)S2=EXP(C*V*T)
17    F3(I)=(U*S2-U-C*V+C*V*S1)/(V*X)
      CALL VCONVO(F3,B,N,N,IWK)
      COMPUTE ESTIMATE OF OT
      DO 20 I=1,N
      FF=F3(I)*H
      WRITE(1)FF
      B(I)=F2(I,2)
      T=(I-1)*H
      S1=0.
      IF(U*T.LT.200.)S1=EXP(-U*T)
20    F3(I)=S1
      CALL VCONVO(F3,B,N,N,IWK)
```

```
      B(1,9)=F2(1,2)
C     COMPUTE PARTIAL OF OT ESTIMATE WRT KGAIN
      DO 25 I=1,N
      T=(I-1)*H
      FF=-F3(I)*H+F2(I)
      B(I,7)=1.
      B(I,8)=FF
      S1=0.
      IF(U*T.LT.200.)S1=EXP(-U*T)
      F3(I)=T*S1
      B(I)=F2(I,2)
      IF(I.EQ.1)GO TO 25
      B(I,9)=(B(I,8)-B(I-1,8))/H
25    CONTINUE
      CALL VCONVO(F3,B,N,N,IWK)
      B(1,6)=0.
      DO 27 I=1,N
      B(I,5)=F3(I)*H
      IF(I.EQ.1)GO TO 27
      B(I,6)=(B(I,5)-B(I-1,5))/H
27    CONTINUE
C     COMPUTE PARTIAL DERIVATIVES OF VARIANCE WRT ALPHA(J)
      DO 40 J=1,3
      DO 35 I=1,N
      T=(I-1)*H
      II=J+6
      B(I)=B(I,II)*B(I,II)
      S1=S2=100.
      IF(U*T.GT.200)S1=0.
      IF(C*V*T.LT.-200.)S2=0.
      IF(S1.NE.0.)S1=EXP(-U*T)
      IF(S2.NE.0.)S2=EXP(C*V*T)
      F3(I)=(C*(S2*(X+C*U*Y)-U*C*Y*S1)/X)**2
35    CONTINUE
      CALL VCONVO(F3,B,N,N,IWK)
      DO 37 I=1,N
      FF=F3(I)*H
37    WRITE(1)FF
40    CONTINUE
C     COMPUTE PARTIAL DERIVATIVE OF VARIANCE WRT KGIAN
      DO 50 I=1,N
      T=(I-1)*H
      S1=S2=100.
      IF(U*T.GT.200)S1=0.
      IF(C*V*T.LT.-200.)S2=0.
      IF(S1.NE.0.)S1=EXP(-U*T)
      IF(S2.NE.0.)S2=EXP(C*V*T)
      B(I,7)=S2*(U*C*Y+X)-U*C*Y*S1
      B(I,4)=A1+A2*B(I,8)*B(I,8)+A3*B(I,9)*B(I,9)
      B(I)=B(I,4)
50    F3(I)=2.*C*C*B(I,7)*(X*(S2*X1-C*Y*S1*(1-U*T))-B(I,7))/(X*X*X)
      CALL VCONVO(F3,B,N,N,IWK)
      DO 53 I=1,N
```

```
      T=(I-1)*H
      B(I,3)=F3(I)*H
      B(I)=A2*B(I,6)*B(I,5)+A3*B(I,5)*B(I,6)
53    F3(I)=2.*C*C*B(I,7)*B(I,7)/(X*X)
      CALL VCONVO(F3,B,N,N,IWK)
C     COMPUTE PARTIAL DERIVATIVE OF VARIANCE WRT GAMMA1
      DO 55 I=1,N
      FF=F3(I)*H+B(I,3)
      WRITE(1)FF
      T=(I-1)*H
      S1=100
      IF(C*V*T.LT.-200.)S1=0.
      IF(S1.NE.0.)S1=EXP(C*V*T)
      B(I)=B(I,4)
      F3(I)=C*C*C*2.*B(I,7)*(S1*(T*(X+U*C*Y)+1)*X-B(I,7))/(X*X*X)
55    CONTINUE
      CALL VCONVO(F3,B,N,N,IWK)
C     COMPUTE PARTIAL DERIVATIVE OF VARIANCE WRT GAMMA2
      DO 60 I=1,N
      T=(I-1)*H
      FF=F3(I)*H
      WRITE(1)FF
      S1=S2=100.
      IF(U*T.GT.200.)S1=0.
      IF(C*V*T.LT.-200.)S2=0.
      IF(S1.NE.0.)S1=EXP(-U*T)
      IF(S2.NE.0.)S2=EXP(C*V*T)
      B(I)=B(I,4)
      F3(I)=C*C*C*2.*U*B(I,7)*(S2-S1)/(X*X)
60    CONTINUE
      CALL VCONVO(F3,B,N,N,IWK)
C     COMPUTE MEAN TRACKING ERROR
      DO 65 I=1,N
      T=(I-1)*H
      FF=F3(I)*H
      WRITE(1)FF
      B(I)=F2(I,2)
      S1=S2=100.
      IF(U*T.GT.200.)S1=0.
      IF(C*V*T.LT.-200.)S2=0.
      IF(S1.NE.0.)S1=EXP(-U*T)
      IF(S2.NE.0.)S2=EXP(C*V*T)
      F3(I)=((U*X1+C*V)*S2-X*X1+C*C*V*Y*S1)/(C*V*X)
65    CONTINUE
      CALL VCONVO(F3,B,N,N,IWK)
      REWIND 1
      DO 75 I=7,9
      DO 75 J=1,N
75    READ(1)B(J,I)
      DO 77 I=4,6
      DO 77 J=1,N
77    READ(1)B(J,I)
      DO 80 I=1,3
```

```
      DO 80 J=1,N
80    READ(1)B(J,I)
      RJ=0
C     COMPUTE STANDARD DEVIATION OF TRACKING ERROR
      DO 83 I=1,N
      F3(I,2)=SQRT(B(I,4)*A1+B(I,5)*A2+B(I,6)*A3)
      DO 82 J=1,6
82    B(I,J)=.5*B(I,J)/F3(I,2)
83    CONTINUE
C     CALUCATE VALUE OF COST FUNCTION
      DO 85 I=N1,N
      F3(I)=F3(I)*H
85    RJ=RJ+(F2(I,3)-F3(I))**2+CE*(F2(I,4)-F3(I,2))**2
      END
      SUBROUTINE GMINV(NR,NC,A,U,MF,MT)
      DIMENSION A(1),U(1),S(10),IP(10)
      COMMON/MAIN1/ NDIM,NDIM1
      TOL=1.E-12
      ADV=TOL*TOL
      MR=NC
      NRM1=NR-1
      NCM1=NC-1
      JJ=1
      DO 5 J=1,NC
      S(J)=DOT(NR,A(JJ),A(JJ))
    5 JJ=JJ+NDIM
      II=1
      DO 19 I=1,NCM1
      XMAX=S(I)
      IMAX=I
      IF(NC.EQ.1) GO TO 15
      IP1=I+1
      DO 10 J=IP1,NC
      IF(XMAX.GE.S(J)) GO TO 10
      IMAX=J
      XMAX=S(J)
   10 CONTINUE
      IF(IMAX.EQ.I) GO TO 15
      S(IMAX)=S(I)
      S(I)=XMAX
      KK=(IMAX-1)*NDIM+1
      CALL SWAP (A(KK),A(II),NR,1)
   15 IP(I)=IMAX
   19 II=II+NDIM
      TOL1=S(1)*ADV
      ADV=TOL1
      JJ=1
      DO 100 J=1,NC
      FAC=S(J)
      JM1=J-1
      JPM=JJ+NRM1
      JCM=JJ+JM1
      DO 20 I=JJ,JCM
```

```
 20 U(I)=0.
    U(JCM)=1.0
    IF(J.EQ.1) GO TO 54
    KK=1
    DO 30 K=1,JM1
    IF(S(K).EQ.1.0) GO TO 30
    TEMP=-DOT(NR,A(JJ),A(KK))
    CALL VADD(K,TEMP,U(JJ),U(KK))
 30 KK=KK+NDIM
    DO 50 L=1,2
    KK=1
    DO 50 K=1,JM1
    IF(S(K).EQ.0.) GO TO 50
    TEMP=-DOT(NR,A(JJ),A(KK))
    CALL VADD(NR,TEMP,A(JJ),A(KK))
    CALL VADD(K,TEMP,U(JJ),U(KK))
 50 KK=KK+NDIM
    TOL1=TOL*FAC+ADV
    FAC=DOT(NR,A(JJ),A(JJ))
 54 IF(FAC.GT.TOL1) GO TO 70
    DO 55 I=JJ,JRM
 55 A(I)=0.
    S(J)=0.
    KK=1
    DO 65 K=1,JM1
    IF(S(K).EQ.0.) GO TO 65
    TEMP=-DOT(K,U(KK),U(JJ))
    CALL VADD(NR,TEMP,A(JJ),A(KK))
 65 KK=KK+NDIM
    FAC=DOT(J,U(JJ),U(JJ))
    MR=MR-1
    GO TO 75
 70 S(J)=1.0
    KK=1
    DO 72 K=1,JM1
    IF(S(K).EQ.1.) GO TO 72
    TEMP=-DOT(NR,A(JJ),A(KK))
    CALL VADD(K,TEMP,U(JJ),U(KK))
 72 KK=KK+NDIM
 75 FAC=1./SQRT(FAC)
    DO 80 I=JJ,JRM
 80 A(I)=A(I)*FAC
    DO 85 I=JJ,JCM
 85 U(I)=U(I)*FAC
100 JJ=JJ+NDIM
    IF(MR.EQ.NR.OR.MR.EQ.NC) GO TO 120
    IF(MT.NE.0)PRINT110,NR,NC,MR
110 FORMAT(I3,1HX,I2,8H M: RANK,I2)
120 NEND=NC*NDIM
    JJ=1
    DO 135 J=1,NC
    DO 125 I=1,MR
    II=I-J+JJ
```

```
125 S(I)=DOT2(NEND,U(JJ),A(II))
    NEND=NEND-NDIM
    II=J
    DO 130 I=1,NR
    U(II)=S(I)
130 II=II+NDIM
135 JJ=JJ+NDIM1
    DO 99 J=2,NC
    K=NC-J+1
    KK=IP(K)
    IF(K.NE.IP(K)) CALL SWAP(U(K),U(KK),NR,NDIM)
 99 CONTINUE
    RETURN
    END
    SUBROUTINE VADD(N,C1,A,B)
    DIMENSION A(1),B(1)
    DO 1 I=1,N
  1 A(I)=A(I)+C1*B(I)
    RETURN
    END
    FUNCTION DOT(NR,A,B)
    DOUBLE PRECISION DOT1,DBLE
    DIMENSION A(1),B(1)
    DOT1=0.D0
    IF(NR.LE.0) GO TO 2
    DO 1 I=1,NR
  1 DOT1=DOT1+DBLE(A(I)*B(I))
  2 DOT=DOT1
    RETURN
    END
    SUBROUTINE SWAP(A,B,N,INC)
    DIMENSION A(1),B(1)
    NN=N*INC
    I=1
  1 IF(I.GT.NN) RETURN
    TEMP=A(I)
    A(I)=B(I)
    B(I)=TEMP
    I=I+INC
    GO TO 1
    END
    FUNCTION DOT2(NN,A,B)
    DOUBLE PRECISION DOT2,DBLE
    DIMENSION A(1),B(1)
    COMMON/MAIN1/NDIM
    DDT2=0.D0
    IF(NN.LE.0) GO TO 2
    DO 1 I=1,NN,NDIM
  1 DDT2=DDT2+DBLE(A(I)*B(I))
  2 DOT2=DDT2
    RETURN
    END
```

```
2
1
100 200 .01
5. -2. -2. .1 .1 .1
1
100 200 .01
10. -6. -6. .001 .01 .01
1
100 200 .01
2. -10. -5. .001 .01 .01
1
100 200 .01
1. -1. -1. .001 .001 .001
1
100 200 .01
1. -4. -4. .0001 .0001 .0001
0
```

APPENDIX D

PROGRAM LISTING OF COMPUTER SIMULATION OF AAA TRACKING TASK

```
      PROGRAM MTQ(INPUT,OUTPUT,TAPE2,TAPE6)
      DIMENSION P(8,2)
      COMMON/VAR/DEL,N,N1,OTDD,EDD,ET,T,C70
      PRINT*,30HUSE EVERY --- SAMPLES?
      REWIND 2
      READ*,I1
      C70=0.
      IC1=0
      DEL=I1*.033
      I5=30/I1
      N=3$N1=9
      PRINT*,40HTYPE IN 1 FOR AZ, 2 FOR EL OR 3 FOR BOTH
      READ*,IFG
      I2=1
      IF(IFG.EQ.2)I2=2
      P(7)=1.28$P(7,2)=1.34
      P(8)=P(8,2)=0.
      DO 15 IC=I2,2
      PRINT*,40HTYPE IN GAMMA1,GAMMA2,KGAIN,P1,P2,P3 FOR,IC
15    READ*,(P(J,IC),J=1,6)
      DO 100 I=1,1050
      READ(2,3)A,B,OTDD,ET,C,EDD
      KK=I+I1-1
      IF(MOD(KK,I1).NE.0)GO TO 100
      IC1=IC1+1
      T=(IC1-1)*DEL
      CALL OBSEL(ELERR,SDEL,C7,P(9))
      IF(IFG.NE.2)CALL OBSAZ(AZERR,SDAZ,C7,P)
      WRITE(6,*)T,AZERR,ELERR,SDAZ,SDEL
      IF(MOD(I,I5).EQ.0)PRINT5,T,AZERR,ELERR,SDAZ,SDEL
      IF(IC1.EQ.1)PRINT5,T,AZERR,ELERR,SDAZ,SDEL
100   CONTINUE
3     FORMAT(6G12.5)
5     FORMAT(5G10.3)
      END
      SUBROUTINE OBSEL(ELERR,SDEL,C7,P)
      COMMON/VAR/DEL,N,N1,OTDD,EDD,ET,T
      DIMENSION A(3,3),Z(3),XX(3,3),R1(3),W1(3,3),W2(3,3),P(8)
      DIMENSION F2(3),GAM1P(3)
      IF(T.GT..001)GO TO 1
      Z(1)=-P(7)
      Z(2)=Z(3)=P(3)*P(7)
      DO 10 J=1,N1
      A(J)=0.
10    XX(J)=0.
      A(1)=P(3)+P(7)*(P(1)+P(3)*P(2))
      A(2)=-P(3)*P(3)-P(7)*P(3)*(P(1)+P(3)*P(2))
      A(4)=1+P(7)*P(2)
      A(5)=-P(3)-P(7)*P(3)*P(2)
      A(7)=-P(7)*P(2)
```

```
      A(8)=P(7)*P(3)*P(2)
      A(9)=-P(3)
      CALL DSCRT(N,A,DEL,W1,W2,10)
      F2(1)=P(7)
      F2(2)=-P(7)*P(3)
      F2(3)=0.
      DO 36 K=1,N
      GAM1P(K)=0.
      DO 36 J=1,N
   36 GAM1P(K)=GAM1P(K)+W2(K,J)*F2(J)
      DO 50 I=1,N
      II=1
      R1(I)=0.
      DO 45 J=I,N1,N
      R1(I)=R1(I)+W2(J)*Z(II)
45    II=II+1
50    CONTINUE
      DO 60 I=1,N
      Z(I)=0.
      DO 60 J=1,N
60    A(I,J)=R1(I)*R1(J)
      DO 20 I=1,N
      I3=I+N
      I2=I3+N
20    R1(I)=W2(I3)+W2(I2)
      Z(1)=Z(2)=Z(3)=0.
      XX(1)=.0001
1     P4=Z(2)-Z(3)+P(3)*Z(1)
      P5=(P4-P(8))/DEL
      IF(T.EQ.0.)P5=EDD
      V=(P(4)+P(5)*P4*P4+P(6)*P5*P5)/DEL
      P(8)=P4
      DO 25 I=1,N
      II=1
      W2(I)=0.
      DO 15 J=I,N1,N
      W2(I)=W2(I)+W1(J)*Z(II)
15    II=II+1
25    CONTINUE
      DO 35 I=1,N
      Z(I)=W2(I)+R1(I)*EDD
35    CONTINUE
      UBAR=-P(1)*Z(1)-P(2)*(Z(2)+P(3)*Z(1)-Z(3))
      IF(ABS(UBAR).LE..785) GO TO 38
      UFIX=.785
      IF(UBAR.GT.0.) UFIX=-.785
      DO 37 J=1,N
   37 Z(J)=Z(J)+GAM1P(J)*(UBAR+UFIX)
   38 CONTINUE
      CALL MULT(W1,XX,N,N1,W2,10)
      DO 40 I=1,N1
40    XX(I)=A(I)*V+W2(I)
      SDEL=SQRT(XX(1))*180/3.14156
```

```
      C7=ET-Z(1)
      ELERR=Z(1)*180./3.14156
      END
      SUBROUTINE OESAZ(AZERR,SOAZ,C7,P)
      DIMENSION AA(3,3),Z1(3),X(3,3),R2(3),WK(3),W3(3,3),W4(3,3)
      DIMENSION GAM1P(3),P(8),F2(3)
      COMMON/VAR/DEL,N,N1,OTDD,EDD,ET,T,C7O
      IF(T.GT..001)GO TO 1
      R2(1)=-P(7)
      R2(2)=R2(3)=P(7)*P(3)
      DO 10 J=1,N1
      AA(J)=0.
10    X(J)=0.
      Z1(1)=Z1(2)=Z1(3)=0.
      X(1)=.00005
      Z1(3)=0.
1     P4=Z1(2)-Z1(3)+P(3)*Z1(1)
      P5=(P4-P(8))/DEL
      IF(T.EQ.0.)P5=OTDD
      V=(P(4)+P(5)*P4*P4+P(6)*P5*P5)/DEL
      P(8)=P4
      C7D=(C7-C7O)/DEL
      D1=COS(C7)
      D2=-TAN(C7)*C7D
      C7O=C7
      AA(1)=D2+D1*P(3)+D1*P(7)*(P(1)+P(2)*P(3))
      AA(2)=-D1*P(3)*P(3)-P(3)*D2-P(7)*P(3)*D1*(P(1)+P(2)*P(3))
      AA(4)=D1+D1*P(7)*P(2)
      AA(5)=-D1*P(3)-P(7)*P(2)*P(3)*D1
      AA(7)=-P(7)*D1*P(2)
      AA(8)=P(7)*P(2)*P(3)*D1
      AA(9)=-P(3)*D1
      CALL DSCRT(N,AA,DEL,W3,W4,10)
      DO 25 I=1,N
      II=1
      WK(I)=0.
      DO 15 J=I,N1,N
      WK(I)=WK(I)+W3(J)*Z1(II)
15    II=II+1
25    CONTINUE
      DO 35 I=1,N
      Z1(I)=WK(I)+(W4(I+3)+W4(I+6))*OTDD
35    CONTINUE
      UBAR=-P(1)*Z1(1)-P(2)*(Z1(2)+P(3)*Z1(1)-Z1(3))
      IF(ABS(UBAR).LE..829) GO TO 38
      F2(1)=P(7)*D1
      F2(2)=-P(7)*P(3)*D1
      F2(3)=0.
      DO 36 K=1,N
      GAM1P(K)=0.
      DO 36 J=1,N
```

```
   36 GAM1P(K)=GAM1P(K)+W4(K,J)*F2(J)
      UFIX=.829
      IF(UBAR.GT.0.) UFIX=-.829
      DO 37 J=1,N
   37 Z1(J)=Z1(J)+GAM1P(J)*(UBAR+UFIX)
   38 CONTINUE
      DO 50 I=1,N
      II=1
      WK(I)=0
      DO 40 J=I,N1,N
      WK(I)=WK(I)+W4(J)*W2(II)*D1
40    II=II+1
50    CONTINUE
      CALL MULT(W3,X,N,N1,W4,10)
      DO 60 I=1,N
      DO 60 J=1,N
      II=(J-1)*N+I
60    X(II)=WK(I)*WK(J)*V+W4(II)
      SDAZ=SQRT(X(1))*180./(3.14156*D1)
      AZERR=Z1(1)*180./(3.14156*D1)
      END
      SUBROUTINE MULT(E,F,L,L1,H,MR)
      DIMENSION E(L1),F(L1),G(9),H(L1)
      DO 10 I=1,L
      II=1
      DO 10 K=I,L
      TEMP=0.
      DO 5 J=I,L1,L
      TEMP=TEMP+E(J)*F(II)
5     II=II+1
      KK=(K-1)*L+I
      H(KK)=TEMP
10    G(KK)=TEMP
      IF(MR.EQ.1)RETURN
      DO 20 I=1,L
      DO 20 K=I,L
      TEMP=0.
      II=K
      DO 15 J=I,L1,L
      TEMP=TEMP+G(J)*E(II)
15    II=II+L
      KK=(K-1)*L+I
20    H(KK)=TEMP
      L2=L-1
      DO 30 I=1,L2
      L3=I+1
      DO 30 J=L3,L
      K1=(I-1)*L+J
      K2=(J-1)*L+I
30    H(K1)=H(K2)
      END
```

```
      SUBROUTINE DSCRT(NDIM,A,DEL,EA,EAINT,NT)
      DIMENSION A(1),EA(1),EAINT(1),COEF(30)
C        SETS EA=EXP(A*DEL),EAINT=INTEGRAL EA 0 TO DEL
      NDIM1=NDIM+1
      NN=NDIM*NDIM
      NTM1=NT-1
      COEF(NT)=1.
      DO 10 I=1,NTM1
      II=NT-I
   10 COEF(II)=DEL*COEF(II+1)/FLOAT(I)
C           NT MUST BE AT LEAST 3
      CALL DIAG(NDIM,EAINT,A,COEF(1),COEF(2))
      DO 60 L=3,NT
      CALL MULT(A,EAINT,NDIM,NN,EA,1)
      IF(L.EQ.NT)GO TO 70
   60 CALL DIAG(NDIM,EAINT,EA,1.0,COEF(L))
   70 DO 80 II=1,NN,NDIM1
      EA(II)=EA(II)+1.0
   80 CONTINUE
      END
      SUBROUTINE DIAG(NDIM,A,B,C1,C2)
      DIMENSION A(1),B(1)
      NDIM1=NDIM+1
      NN=NDIM*NDIM
      NM1=NDIM-1
      II=1
      IF(C1.EQ.1.0) GO TO 10
      DO 5 J=1,NN,NDIM
      K=J+NM1
      DO 4 I=J,K
    4 A(I)=C1*B(I)
      A(II)=A(II)+C2
    5 II=II+NDIM1
      RETURN
   10 DO 7 J=1,NN,NDIM
      K=J+NM1
      DO 6 I=J,K
    6 A(I)=B(I)
      A(II)=A(II)+C2
    7 II=II+NDIM1
      RETURN
      END
```

REFERENCES

[1] D.G. Luenberger, "An Introduction to Observers," IEEE Transactions on Automatic Control, Vol. AC-16, pp 596-602, December 1971.

[2] R. S. Kou and B. C. Glass, Development of Observer Moder for AAA Tracker Response, AMRL-TR-79-77 (in press), Aerospace Medical Research Laboratory, Wright-Patterson AFB, Ohio, August 1979.

[3] M. M. Vikmanis and R. S. Kou, Monte Carlo Simulation of an AAA System Using the Observer Model, AMRL-TR-79-78 (in press), Aerospace Medical Research Laboratory, Wright-Patterson AFB, Ohio, August 1979.

[4] R. S. Kou, B. C. Glass, and M. M. Vikmanis, Reduced-Order Observer Model for AAA Tracker Response, AMRL-TR-79-79 (in press), Aerospace Medical Research Laboratory, Wright-Patterson AFB, Ohio, August 1979.

[5] R.S. Kou, B.C. Glass, C.N. Day, and M.M. Vikmanis, "AAA Gunner Model Based on Observer Theory," Proceedings of the 14th Annual Conference on Manual Control, Los Angeles, CA, April 25-27, 1978.

[6] R.S. Kou, B.C. Glass, C.N.Day, and M.M. Vikmanis, "Development of an AAA Gunner Model Based on Reduced-Order Observer Theory" to be presented at the 17th IEEE Conference on Decision and Control at San Diego, CA, January 10-12, 1979.

[7] R.S. Kou, B.C. Glass, M.M. Vikmanis, M.S. Moran, and C.N. Day, "A Linear Time-Varying Antiaircraft Gunner Model Based on Observer Theory," to be presented at the 1978 International Conference on Cybernetics and Society at Tokyo, Japan, November 3-7, 1978.

[8] R.S. Kou, B.C. Glass, M.S. Moran, C.N. Day, and W.C. Summers, "Parameter Identification of an Antiaircraft Gunner Model," to be submitted to the AIAA Journal of Guidance and Control.

[9] D.L. Kleinman, S. Baron, and W.H. Levison, "A Control Theoretic Approach to Manned-Vehicle Systems Analysis," IEEE Trans. on Automatic Control, Vol. AC-16, No. 6, December 1971.

[10] E.P. Rolek, "MTQ Experimentation Report No. 5-MTQ AZ EL One Study Initial Analysis," Systems Research Laboratories, Inc., Report No. 5874-15-18, September 1977.

[11] A.P. Sage and J.L. Melsa, System Identification, New York, Academic Press, 1971.

[12] P. Eykhoff, System Identification, New York, John Wiley & Sons, 1974, p. 161.

☆U.S.Government Printing Office: 1979 — 657-084/393